

No. 1608  
\$12.95

# **A KID'S MANUAL FOR PROGRAMMING THE SINCLAIR/TIMEX COMPUTERS**

**BY ED HOORNAERT**



**TAB BOOKS Inc.**

BLUE RIDGE SUMMIT, PA. 17214

FIRST EDITION

FIRST PRINTING

Copyright © 1983 by TAB BOOKS Inc.

Printed in the United States of America

Cartoon illustrations by Diane Mainer.

Reproduction or publication of the content in any manner, without express permission of the publisher, is prohibited. No liability is assumed with respect to the use of the information herein.

Library of Congress Cataloging in Publication Data

Hoornaert, Ed.

A kid's manual for programming the Sinclair/Timex computers.

Includes index.

1. Timex 1000 (Computer)—Programming. 2. Timex Sinclair 2000 (Computer)—Programming. I. Title.  
QA76.8.T48H66 1983 001.64'2 83-4870

ISBN 0-8306-0108-2

ISBN 0-8306-0608-4 (pbk.)

Cover photograph by the Ziegler Photography Studio of Waynesboro, PA.

# Contents

---

<b>1</b>	<b>Speech Lessons</b>	<b>1</b>
	Introductions • PRINT • ENTER • Quotation Marks • DELETE • SHIFT • Red Characters • Syntax Errors • Your First Program • RUN • NEW • Program Bugs • LOAD • Learning the Keyboard • Line Numbers • Using What You've Learned • Quiz 1	
<b>2</b>	<b>Pretty Pictures, Some That Move</b>	<b>20</b>
	CLS • Computer Modes • GRAPHICS • Inverse Video • Graphics Characters • Graphics Demonstration • Using Graphics Characters • SAVE • Graphics Pad • PRINT AT • FUNCTION • GOTO • PAUSE • BREAK • Using What You've Learned • Quiz 2	
<b>3</b>	<b>Crunching Numbers</b>	<b>44</b>
	PRINT • Functions SQR, LN, COS, TAN • FAST • SLOW • Variables • LET • E (exponent) • Row and Column Coordinate Practice • GOSUB • RETURN • STOP • Using Subroutines • Using Variables • Using What You've Learned • Quiz 3	
<b>4</b>	<b>Here We Go Loop the Loop</b>	<b>60</b>
	Flowcharts • FOR • TO • NEXT • SCROLL • Using FOR...NEXT Loops • LIST • Keyboard Drill • Keywords • CONT • Semicolon in Print Statements • Comma in Print Statements • Using What You've Learned • Quiz 4	
<b>5</b>	<b>Putting in Input</b>	<b>82</b>
	INPUT • Using INPUT • String Variables • Using String Variables • REM • DIM • Arrays • Program Structure • Random Number • INT • RND • Using RND • STEP • Quiz 5	

<b>6</b>	<b>Decisions, Decisions</b>	<b>105</b>
	Using GOTO • IF, THEN • Using IF...THEN • INKEY\$ • EDIT • Cursor Arrow Keys • Using EDIT to Modify a Program • Writing a Menu • Using INKEY\$ and IF THEN • Using What You've Learned • Quiz 6	
	<b>Insecticide Patrol Answers</b>	<b>127</b>
	<b>Tape Recorder Use</b>	<b>130</b>
	<b>Program Listings</b>	<b>132</b>
	<b>Index</b>	<b>156</b>

## To Parents

---

Is your child ready for the computer age?

In many ways the Sinclair ZX81 and Timex Sinclair 1000 microcomputers are excellent machines on which your child can learn programming. Automatic syntax checking and single key-stroke entry of commands, such as PRINT, are just two of the features that will greatly help your child.

This book is designed as a self-teaching course in the Sinclair version of the BASIC computer language. You need not know BASIC yourself. You will be called upon to make sure that a cassette recorder is correctly connected to the computer when game or quiz programs are to be loaded. See the computer manual, or page 130, for directions.

The quizzes and learning games in the book are an important part of the source. New ideas such as editing are introduced via the games. Skills such as learning the keyboard are also drilled. The games and quizzes are available on a ready-to-run cassette (see page 159) and are also listed in the back of this book if you wish to program them yourself.

Shortly after beginning the course, your child will be writing his or her first program. The most commonly used commands are covered. By the end of the book, your child should be armed with a working knowledge of Sinclair BASIC—and will have entered the computer age.





## Chapter 1

### Speech Lessons

#### LESSON 1

Clara, meet the human reading this book. Human, meet Clara.  
Who, you may ask, is Clara?



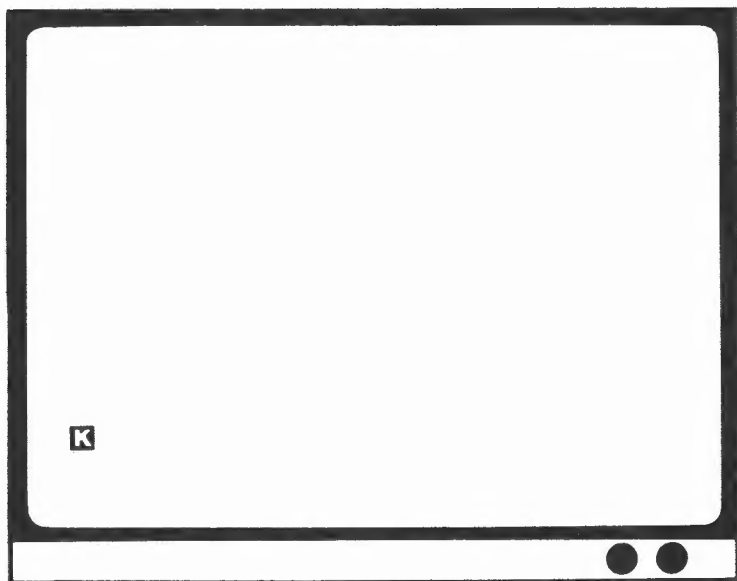
Clara is a computer designed by the Sinclair company and sold under the names of Sinclair ZX81 or Timex Sinclair 1000. By any name, though, Clara is a marvelous machine that can do many, many things, such as math, games, or interesting pictures and patterns. This book shows you how to make Clara do these marvelous things.

To give Clara directions, you must speak computer language. Clara uses a language called BASIC. It's a lot like English, but it doesn't have as many words. This makes BASIC easy to learn. This book will show you how to talk like a computer.

I hope you and Clara have fun together!

## LESSON 2

The computer is plugged in, the TV is on. A black K is at the bottom of the screen.



Now what? How can you make the computer do something?

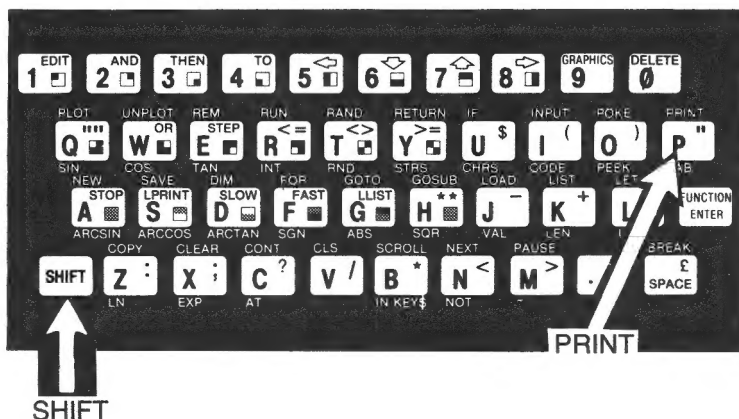
For starters, let's make Clara speak. Well, not really. Maybe someday you'll have a computer that can talk, but Clara can only print—not even write, just print.

1. First, find the word PRINT at the right side of the keyboard.
2. Press the P key underneath PRINT. The word PRINT appears on the screen.
3. Next, find SHIFT



4. Hold down SHIFT while you press the P key again. A quotation mark (") appears on the screen.

PRINT "



5. Now type the letters that spell 'hello'.

PRINT "HELLO

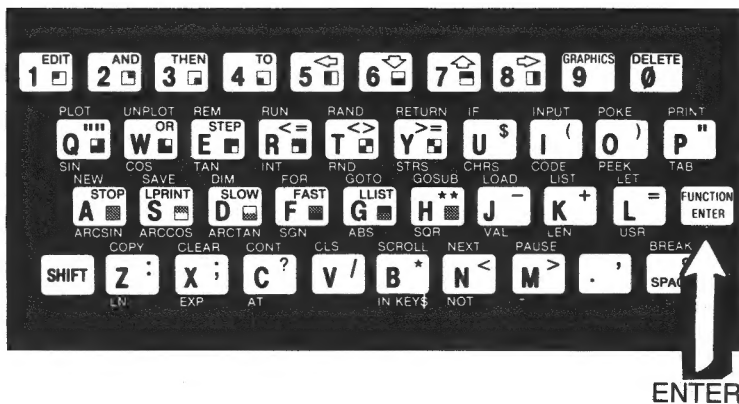
6. Finally, hold down SHIFT while you press P again to print another quotation mark.

PRINT "HELLO"

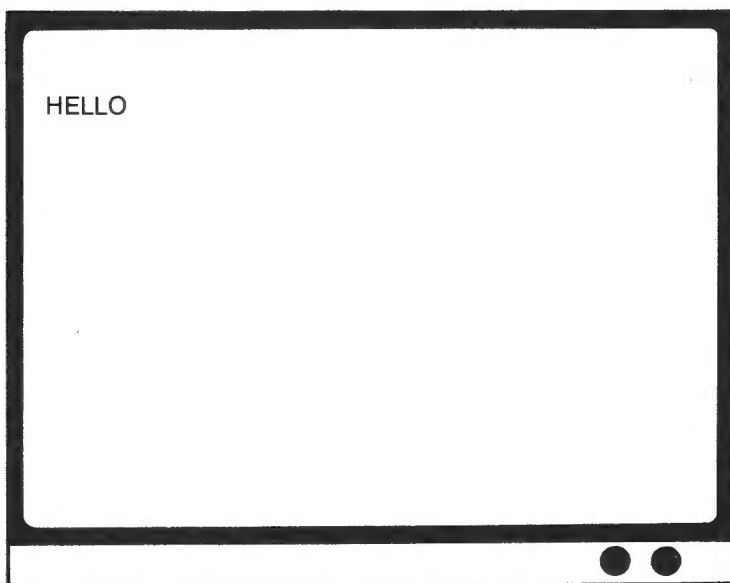
The computer isn't really doing anything. It is ignoring you, paying no attention to what you just typed. What a rude computer!

To make the computer pay attention to you, you must press the key marked ENTER at the right of the keyboard.





Press ENTER. The screen should now look like this.



Congratulations! You have just made that rude computer say a polite 'hello' to you.



**Review** \*\*\*\*\*

1. PRINT—tells the computer to print whatever is inside the quotation marks.

2. Quotation marks (“ ”)—You’ve seen them in books, around words that someone says. On the computer, quotation marks go around words that the computer is to print.

**LESSON 3**

Let’s give Clara some more talking lessons. Type in each of the following. To leave a space between words, press SPACE at the bottom right of the keyboard.

PRINT “HICCUP”

then ENTER

PRINT “BELCH”

then ENTER

PRINT “I AM A COMPUTER”

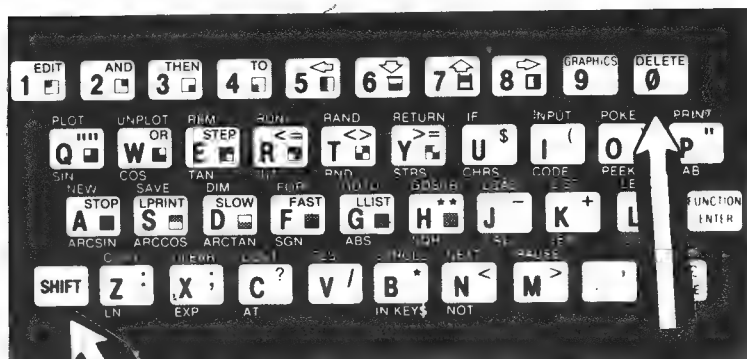
then ENTER

OOPS!

If you goof while typing, you *can* correct it—but only if you haven’t already pressed ENTER.



1. Find DELETE in red letters at the top right of the keyboard.
2. Hold down SHIFT while you press DELETE.



DELETE

SHIFT

This erases, or deletes, the number or letter to the left of the (L). Keep erasing till the mistake is gone, then type it correctly.



The next sentence is a bit different, as it uses a question mark (?) and a comma (,). Can you find them on the keyboard? They are in red letters on the bottom row of keys.

To type question mark, DELETE or any other 'red character' hold down SHIFT while you press the key.

PRINT "WHAT IS YOUR NAME, HUMAN?"  
PRINT "your own name"

then ENTER  
then ENTER

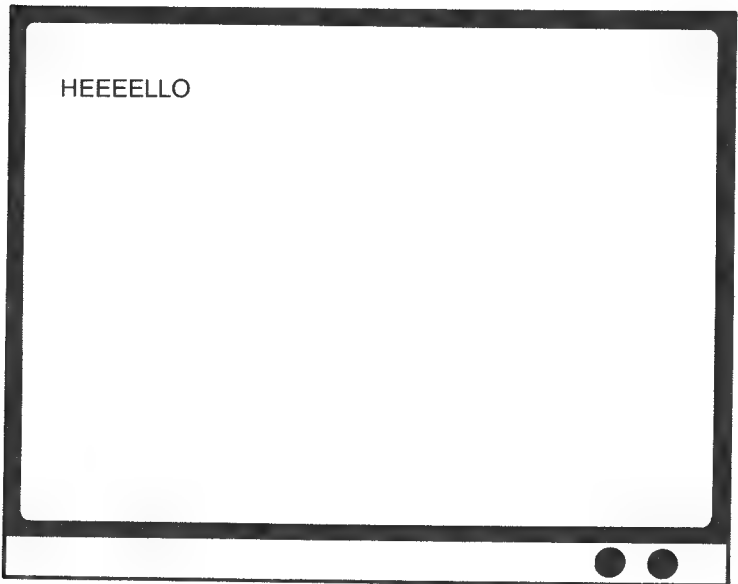
## Review \*\*\*\*\*

1. DELETE—holding down SHIFT while you press this key erases a number or letter, but only before you press ENTER.
2. 'Red characters' (such as " , ? DELETE)—to type these, hold down SHIFT.
3. ENTER—remember to press ENTER at the end of each line.

## LESSON 4

Well, you've taught Clara a new trick—to talk in English. It may seem to you that your computer is pretty smart. Really, though, it is dumb—dumber even than that silly kid in 6th grade (you know who I mean). You don't believe me? Then type the following, exactly the way you see it.

PRINT "HEEELLLO"      then ENTER

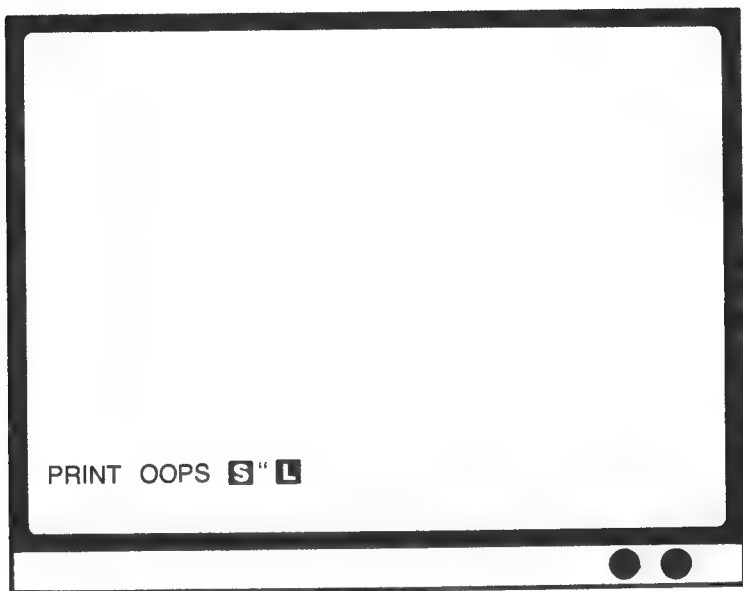


See? the dumb computer doesn't even know how to spell. It just prints whatever you tell it. Can you make the computer say some other silly things? Try it.

Computers are dumb in other ways, too. Unless you give them exactly the right directions, they just can't figure out what you mean. For example, type this—leaving out the first quotation mark.

PRINT OOPS" then ENTER

The dumb computer gets so confused by that missing quotation mark that it can't do anything.



The **S** tells you where something is missing—the first quotation mark. If you erase the line, the **S** disappears. **S** really stands for Syntax Error, but you can think of it as Silly Mistake.

So don't confuse poor Clara—remember the quotation marks!

**Review** \*\*\*\*\*

**S**—means the computer doesn't understand what you want it to do. Check for quotation marks.

## LESSON 5

Everyone likes jokes, right? Maybe we can get Clara to tell jokes. Let's try something. Type this, then ENTER it.

PRINT "TELL A JOKE, CLARA"

TELL A JOKE, CLARA

Well, that didn't work. To make Clara tell a joke, we need to write a program. A program is just a list of directions, telling the computer exactly what to do. So that the computer knows what to do first, the directions must be numbered.



1. Type the number 1. This tells the computer to do this line first.
2. Press PRINT, then quotation mark.
3. Type the first line of our joke, then ENTER it.

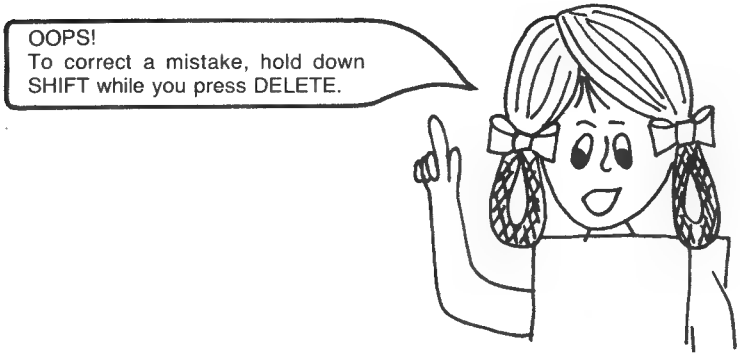
```
1 PRINT "NAME ONE ROOM NO ONE CAN ENTER"
```

Notice that instead of printing just the words of our joke, Clara printed exactly what we typed, including the line number, 1, and the command PRINT. That is how the computer acts when you are programming (writing a program).

Now type in the rest of the joke. Remember the line numbers!

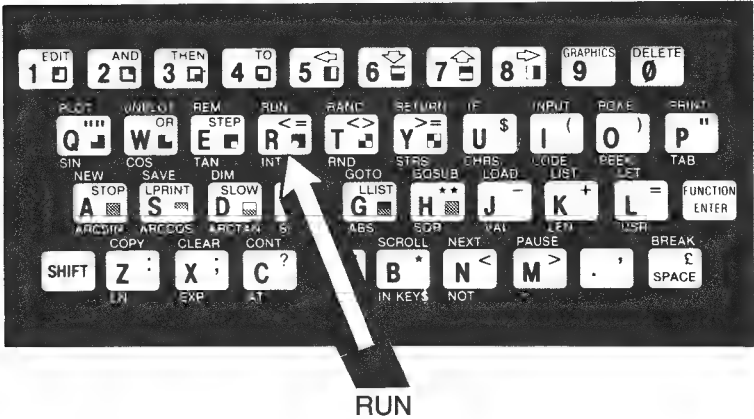
2 PRINT "I GIVE UP"                    then ENTER

3 PRINT "A MUSHROOM"           then ENTER



Our program is done, but it doesn't look very good, does it? The line numbers, PRINTs and quotations mark get in the way. Do you know how to make the computer run our joke program?

- 1. Find the word RUN above the R key.



- 2. Press the R key, below the word RUN.
- 3. Press ENTER.



Presto! You have just run your first program! (If it doesn't work, check your program to see that it is typed correctly. If necessary, retype it.) Run your program several times, using the three steps shown above.

**Review \*\*\*\*\***

1. Program—a list of directions telling the computer what to do.
2. Line numbers—numbers at the beginning of each program line that tell the computer what to do first, second, and so on.
3. RUN—tells the computer to follow the directions in the program that is in its memory.

## LESSON 6

In the last lesson you learned how to write a program. The computer 'remembers' your program by storing it in its memory, just as you remember the things stored in your memory. However, a computer is very different from you in the way it forgets.

Sometimes people get amnesia, perhaps from a bump on the head, and they forget everything—even their own names. Pounding Clara on the head is not a good idea, though.



One way to give the computer amnesia is to unplug the computer. There is a better way, however.

1. Find and then press NEW
2. ENTER



NEW

The command NEW tells the computer to ‘forget’ all the programs and information in its memory. You just gave the computer amnesia! It is a good idea to type NEW every time you begin to type a program so that two programs don’t get jumbled together.

Now that the computer’s memory is blank, try writing a program of your own. Think of a joke that you. Write a program to make the computer tell your joke. Remember your line numbers and quotation marks. Just in case you need it, here is another example to guide you.

- 1 PRINT “WHY ARE FISH SO SMART?”
- 2 PRINT “BECAUSE THEY TRAVEL IN SCHOOLS.”

## Review \*\*\*\*\*

Make sure you understand the words and ideas on this page and the next. You’ll have a computer quiz on them a bit later.

1. PRINT—tells the computer to print whatever is inside the quotation marks.

2. RUN—tells the computer to start following the list of directions, or program, that is in its memory.

3. NEW—tells the computer to forget the program in its memory.

4. SHIFT—you need to hold down the SHIFT key to type any of the red characters on the keys, such as DELETE ? , “

5. Quotation marks (“)—they belong on each side of the words to be printed by the computer.

6. DELETE—this erases the number or letter to the left of the **L**

7. Program—a detailed list of directions telling the computer what to do.

8. Line Numbers—they are put at the beginning of each program line. They tell the computer the order in which to do the steps in the program.

9. **S**—Silly mistake. Make sure that you have all the needed quotation marks.

10. ENTER—at the end of each program line or command, make the computer pay attention to you by pressing ENTER. From now on I won't always remind you to press ENTER.

## INSECTICIDE PATROL #1



Clara hates bugs!

When a program doesn't work properly, or doesn't work at all, we say it has a *bug* in it. Debugging a program means getting rid of the programming error.

This program doesn't work. Can you debug it? (Answer on page 127.)

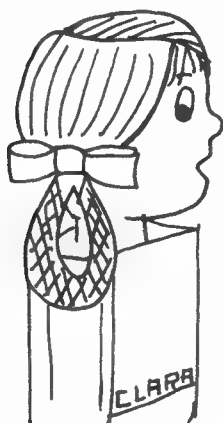
```
10 PRINT "CLARA"  
20 PRINT "HATES"  
30 PRINT "BUGS"
```

## LESSON 7

You've worked hard so far, learning the basics of BASIC. Now for some fun!

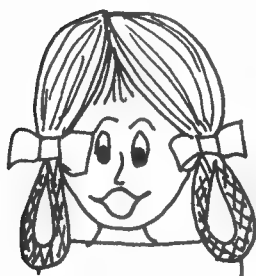
This book has many computer programs for you to enjoy and learn from, but do you know how we can get a program into Clara without having to type it all in?

Computer programs can be stored on cassette tapes, exactly like the cassette tapes you have seen before. Several cassette tapes are available to go with this book (See page 159). Game 1, which you will soon be playing, gives you the chance to pile up a big score while becoming more familiar with the computer keyboard. The more often you play it, the faster your typing will be.



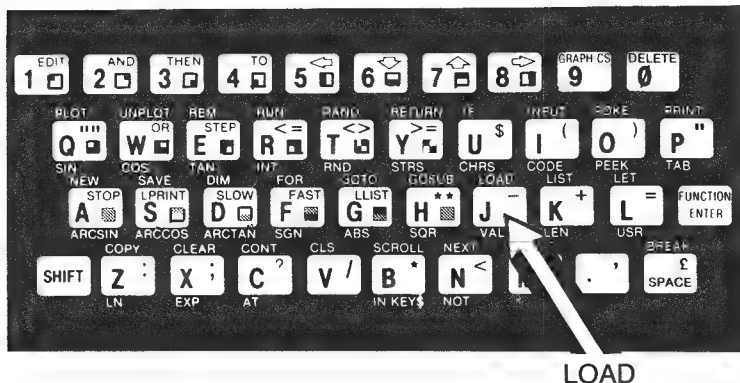
Loading a program from tape can be tricky at first, so have your parent or teacher help you till you get the hang of it.

1. Have an adult make sure the tape recorder is properly connected to the computer and that the volume on the recorder is at the proper level. (See page 130 for more complete directions if you need them.)
2. Find the cassette labelled Chapter 1. (If you don't have the cassette, the program is listed beginning on p. 132.)



Never put a cassette tape near electricity, magnets, heat or water. These things may damage the program.

3. Open the tape recorder by pressing the Eject button.
4. With the title 'Chapter 1' on the top side, gently push the cassette into the recorder.
5. Close the tape recorder.
6. Press 'rewind' on the recorder. When the tape is completely rewound press the 'Stop' button.
7. Press the key underneath the command LOAD. Don't ENTER yet.



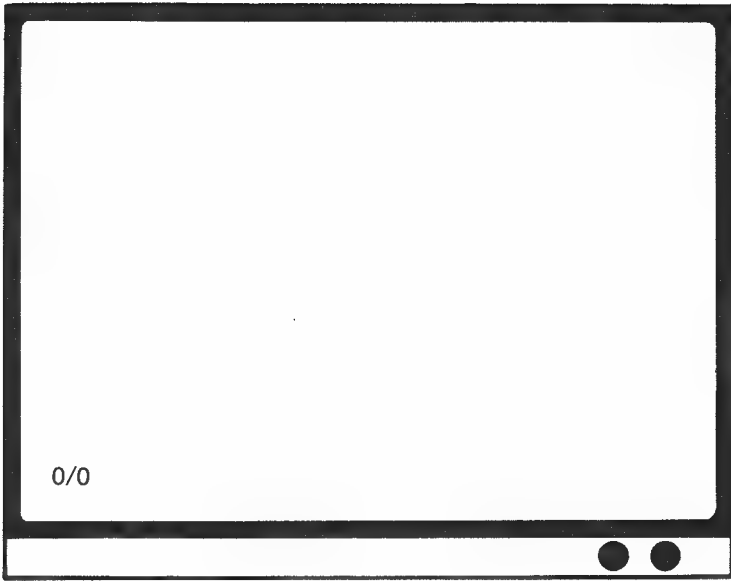
8. Type the name of the program, exactly as follows.

LOAD "GAME 1"      (Don't ENTER yet!)

9. Finally, press ENTER and immediately push the 'Play' button on the tape recorder.



After a short pause, the TV screen will go crazy. Don't worry. That's just what happens when loading Clara. Loading the program should take no more than a minute. Then the screen should look like this:



If it doesn't, press the space key to stop the wavy lines, and get your adult helper. Resetting the volume level a bit higher or a bit lower usually helps, but you might have to experiment to find the correct setting. See page 130 for more details on setting up the tape recorder.

Do you remember how to run a program? That's right—RUN and ENTER. Have a good game! Feel free to play it many times!

## LESSON 8

You know enough already to make programs that do more than just tell jokes. Type NEW to clear any programs from Clara's memory. Study the program at the bottom of the page. Can you figure out what it will do?

Notice that the line numbers count by tens. This is the way big shot computer programmers write their programs, to make it easier to add extra lines if needed. Computers have separate keys for zero - 0 - and the letter o - O. Clara won't understand if you mix up the two, and will show you **S** for Silly mistake.

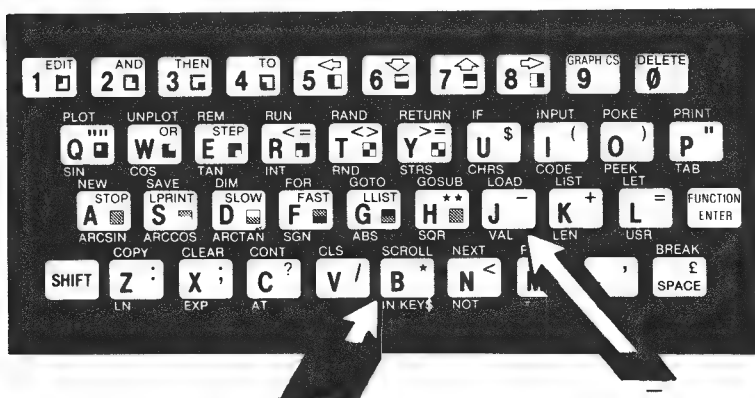
Very carefully, type in this program. The spaces between the letters are important, so I've shown them with periods. When you see a period, type a space.

```

10 PRINT ".*****"
20 PRINT "I.O.O.I"
30 PRINT "I..A..I"
40 PRINT "I---I"
50 PRINT ".V...V"
60 PRINT "..VVV"

```

Here are the locations of the dash and asterisk keys. These are red characters, so remember to hold SHIFT:



Now RUN the program. Can you see a face? This bearded fellow looks rather sad, don't you think?

Now press the ENTER key. Your program appears once more. The computer had it stored in its memory, even though you couldn't see it on the screen. Try running the program again. Can you make Clara show your program a second time? That's correct—press ENTER.

Let's change our picture to give our bearded man hair that sticks up. We need to change line 10. We can do this by writing a new line 10. Type this, and remember that the period stands for a space.

```

10 PRINT ".WWWWW"

```

As soon as you hit ENTER, Clara forgets the old line 10. The W's are now part of our program. Run it a few times.

Can you make some changes of your own? Can you rewrite 20 to give the man different eyes? Can you change line 40 to give him a different mouth?

## LESSON 9

Your turn now! I drew the last picture—you draw this one. Think of a simple picture, like a tree or a square or an animal. Write a program that will draw your picture.

1. Use your imagination, but plan carefully. Don't be discouraged if it doesn't work the first time!
2. Make sure your line numbers count by tens—10, 20, 30, 40, and so on.
3. To change a line, type a new line with the same number.



When you're finished, impress your friends or parents by showing them your program.

Good luck!

## LESSON 10

Quiz time! I know you're just *dying* to have a quiz, but this one is different—it is a computer quiz. Only you and Clara will know how you do. You'll either amaze yourself with your brilliance, or you'll find out what you need to review.

You should get an adult's help to load the quiz program.

1. Put the 'Chapter 1' cassette into the tape recorder and rewind it. (See page 133.)
2. Type LOAD "QUIZ 1".
3. Press ENTER and the Play button. (Look back at Lesson 7 if you need further directions.)



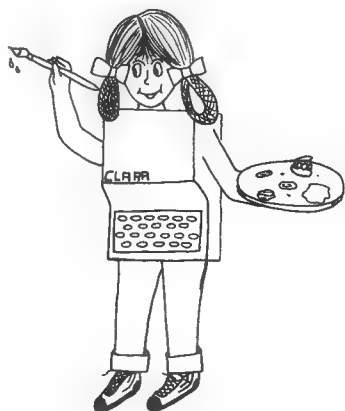
The questions are below, but the possible answers are shown only on the TV screen.

1. Read question 1 below.
2. Find the best answer on the TV screen.
3. Type the *number* of the answer you choose.
4. The computer will tell you if your answer is right or wrong.  
If you were wrong you get a second chance before Clara tells you the right answer.



## QUIZ 1

1. Which BASIC command erases a number or letter?
2. Which symbol shows that you made a Silly mistake, like typing Ø instead of O?
3. Find the quotation marks.
4. What key should you hold down while you type any of the red characters (such as " ? \* DELETE)?
5. Which command tells Clara to show on the screen whatever is in between the quotation marks?
6. Which command tells the computer to start following the directions in the program in its memory?
7. Which command must you use to get a program from a cassette tape into Clara's memory?
8. Which command gives the computer amnesia (that is, makes it forget everything in its memory)?
9. Which key should you press to make the computer pay attention to what you've typed?
10. Find the line numbers.



## Chapter 2 Pretty Pictures, Some That Move

### LESSON 1

So far you've learned the basics of making Clara print words on the screen. Now you'll find out how to make an even more impressive display—a dancing word!



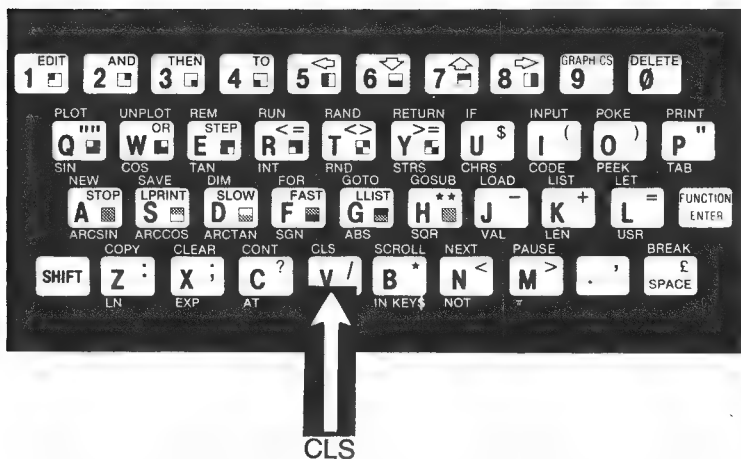
We need a word that is nice and short for this program, because you will be typing it a lot. Your name will do nicely—unless it's Englebert or Murgatroyd, or something else really long. If you have a long time, use a nickname.

First type NEW to clear the computer's memory. Then begin the program by typing your name.

```
10 PRINT "your name"
```

Since we are trying to make your name move around the screen, we need to erase it before we print it again somewhere else. To do this we use the command CLS, which stands for CLear the Screen.

```
20 CLS
```



Now we'll move your name over a bit, then clear the screen again.

```
30 PRINT "    your name"
      (3 spaces)
40 CLS
```

Try running this part of the program, just to see what it is doing. If you looked quickly you could see your name seem to move a bit to the right. To make it move farther, type the following.

```

50 PRINT "        your name"
      (6 spaces)
60 CLS
70 PRINT "          your name"
      (9 spaces)
80 CLS
90 PRINT "            your name"
      (12 spaces)
100 CLS
110 PRINT "              your name"
      (15 spaces)
120 CLS
130 PRINT "                your name"
      (18 spaces)

```

Now run the program.

Can you add more lines to this program to make your name dance back to the left of the screen?

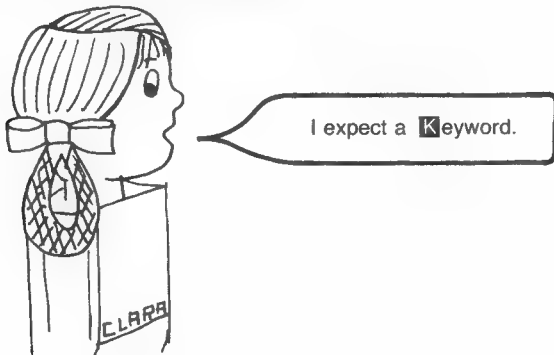
**Review** \*\*\*\*\*

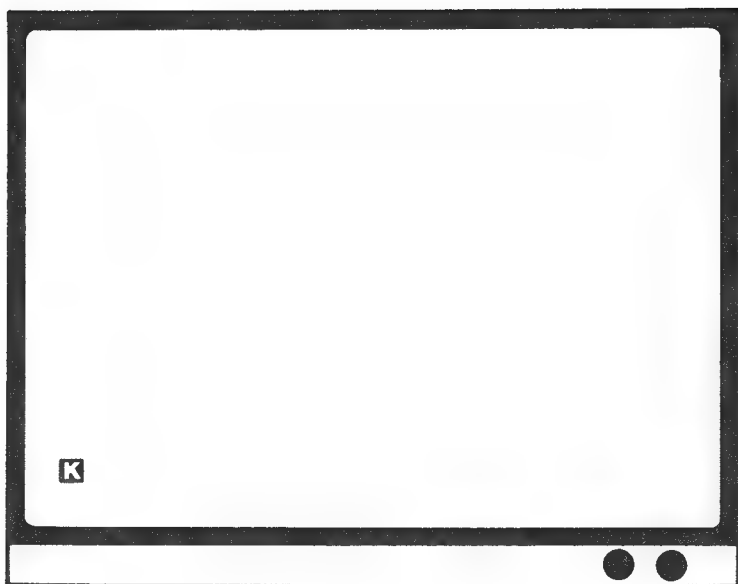
CLS—tells the computer to clear the screen.

## LESSON 2

Have you ever wondered why, when you press the P key, sometimes the screen shows a P and sometimes the word PRINT? Sometimes R and sometimes RUN? Sometimes V and sometimes CLS?

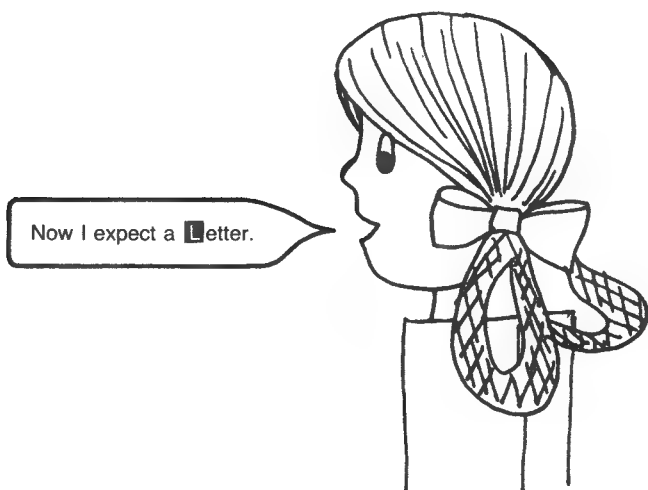
It all depends upon what Clara expects from you. At the beginning of every line, the computer expects a command, or keyword, like PRINT, LOAD, NEW, etc. Clara tells you it expects a keyword by showing a black **K**.

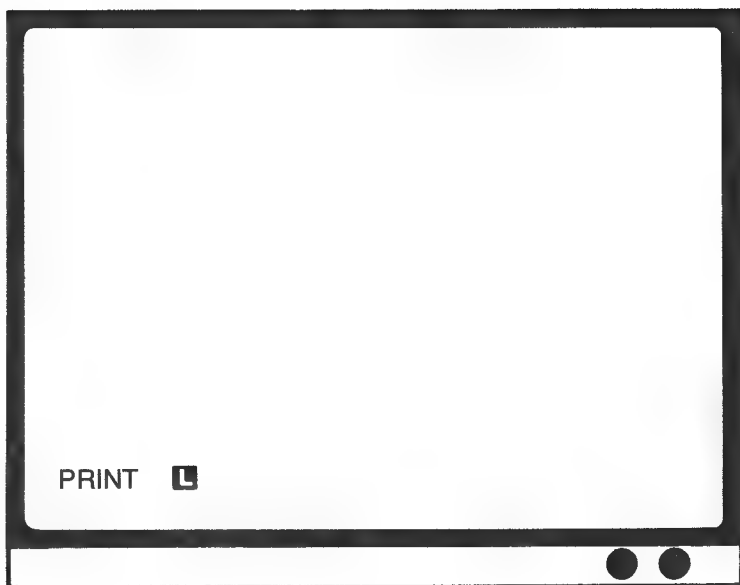




Only line numbers or keywords (the commands written *above* the keys themselves) will be printed when the screen shows **K**.

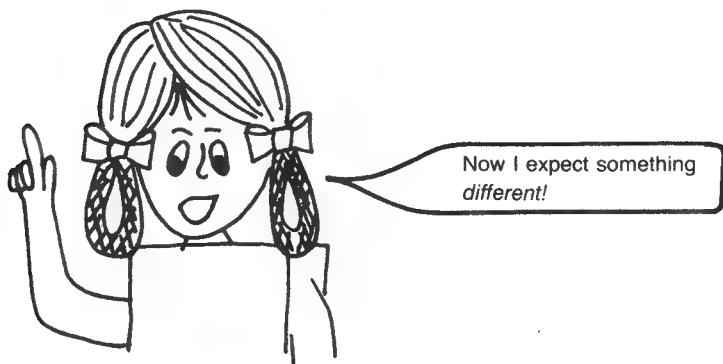
After you have given the computer a command, such as PRINT, it expects something different from you. Clara expects you to type a letter, or a number, or a quotation mark. The computer tells you it expects a letter rather than a keyword by showing a black **L**.

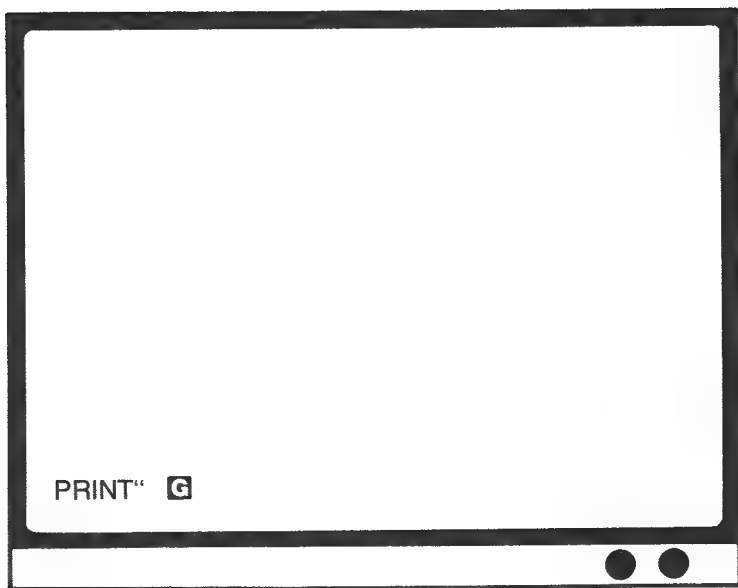




Only letters or numbers or the shifted 'red characters' can be printed when the screen shows **L**.

There is still another kind of mark Clara can show besides (K) and **L**. This mark is **G**. When Clara shows you **G** it expects something different, and it will print something that is *quite* different.

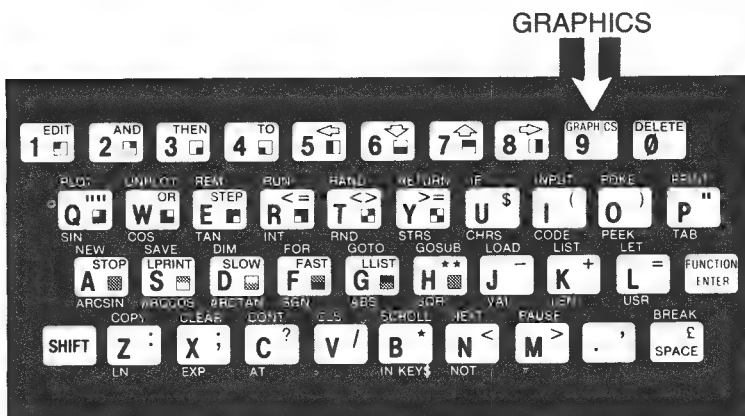




1. To find out what **G** does, first type this.

PRINT "

2. Then find the word GRAPHICS on the keyboard. (Yes, **G** stands for Graphics.)
3. Since GRAPHICS is a red character, hold down SHIFT while you press GRAPHICS.



I'm not going to tell you yet what GRAPHICS does—I'll let you experiment! Two pointers may help you, though.

1. To get out of graphics, hold down SHIFT while you press the GRAPHICS key again.
2. At the end of the PRINT line, remember the final quotation mark.

**Review \*\*\*\*\***

1. **K**—shows that the computer will print only key-words (commands).
2. **L**—shows that the computer will print the letters or characters *on* the keys.
3. **G**—shows that the computer is ready to do graphics (whatever that is).

**LESSON 3**

I'll bet you've figured out what GRAPHICS does! Yes indeed, it prints the letter *in reverse*—white letters on a black background instead of the other way around. Computer bigshots call this *inverse printing*. **P** is inverse P, and so forth.

Inverse printing is great. As an example of how to use it, study the following 'poem' based on the name Chris.

```
C OURAGEOUS
H APPY
R IGH-T-HANDED
I NGENIOUS
S MILEY
```

Do you see how the inverse letters at the beginning of each line spell 'Chris'? A program to print this poem would start like this.

```
10 PRINT "C OURAGEOUS"
20 PRINT "H APPY"
```

To type these lines, begin the usual way.

1. Type the line number 10, the keyword PRINT, and the quotation mark.

```
10 PRINT "
```



- 2 Then hold SHIFT while you press GRAPHICS.
3. Type C, which appears on the screen as **C**.

## 10 PRINT "**C**"

4. To get out of graphics, hold SHIFT while you press the GRAPHICS key again.
5. Type the rest of the word (ourageous) and then the quotation mark.

## 10 PRINT "**C**OURAGEOUS"

Now think up a poem like this about your own name. Make a program that will display your poem, with your name in inverse letters.

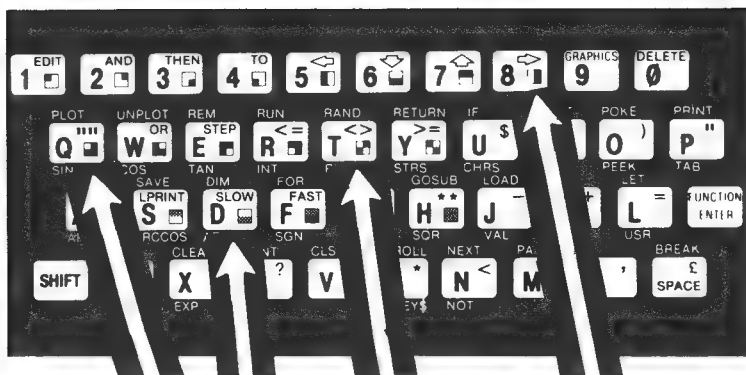
### Review \*\*\*\*\*

Inverse printing—white on black, instead of the other way around. To get inverse printing, first make Clara show **G** by holding SHIFT while you press GRAPHICS. Then type the letter you want.

## LESSON 4

Do you know what the word *graphics* means? According to my dictionary, it means "the art of drawing." Clara can draw interesting pictures using special graphics characters. You've probably noticed them on the keyboard.





## Special Graphics Characters

Play around a bit to see if you can make Clara print any of the graphics characters. If you give up, follow the directions below.

1. Type PRINT and quotation mark.
2. Get into graphics mode (SHIFT and GRAPHICS).
3. Hold down SHIFT while you press a graphics character key, like the Q key. There it is!

PRINT " █

Now look carefully over the keyboard and make Clara print all graphics characters you can find. When you are done you will need a quotation mark at the end of the line. Inverse quotation marks just won't do. (Clara wouldn't understand—Silly mistake again.) You will need to get out of graphics mode, so hold SHIFT while you press the GRAPHICS key. Then type the quotation mark.

\* \* \*

Time for another game!

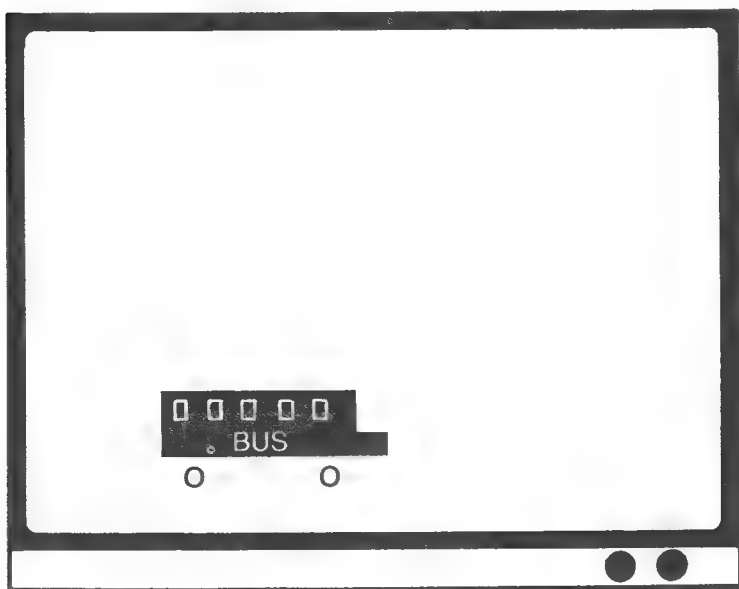
When you go to a restaurant, you choose your food from a menu. Many computer programs let you choose from a menu, too—a menu showing what is in the program. The cassette game for Chapter 2 is 'menu driven', as big shot programmers say. Clara will show you a menu of the graphics games available, and you decide what you want. (No hamburgers on this menu, though. Sorry!)

1. Put the cassette marked Chapter 2 into the tape recorder and rewind. (See page 134.)

2. Make sure that the recorder is properly attached and that the volume level is set.
3. Type LOAD "GAME 2"
4. Press ENTER and the recorder's Play button at the same time.
5. If you need further directions, look back at Lesson 7 in Chapter 1.

## LESSON 5

Having a program like Game 2 that lets you draw is all well and good. But how do you draw things like robots or cars in your own programs? How would you draw this bus, for example?



1. Type the following:

```
10 PRINT "
```

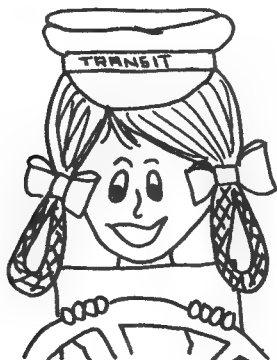
2. Now get Clara set to draw graphics characters by pressing SHIFT and GRAPHICS.
3. Type five of the shape ■ (graphics E) by holding SHIFT and pressing E five times.

10 PRINT " **TTTT**

4. Next it's **█** (graphics 5) so press SHIFT and the 5 key.

10 PRINT " **TTTT**

5. To type the quotation mark needed at the end of this line we need to get out of graphics mode. Press SHIFT and the GRAPHICS key, then add the quotation mark. ENTER.



6. Type this and then get into graphics mode.

20 PRINT " **EUS**

7. Press the space key, B, U, S, space, space and **█** (hold SHIFT and press 5).

20 PRINT " **EUS**

then press ENTER.

For the last line of our program, we don't need to use graphics at all.

1. Type this:

30 PRINT "

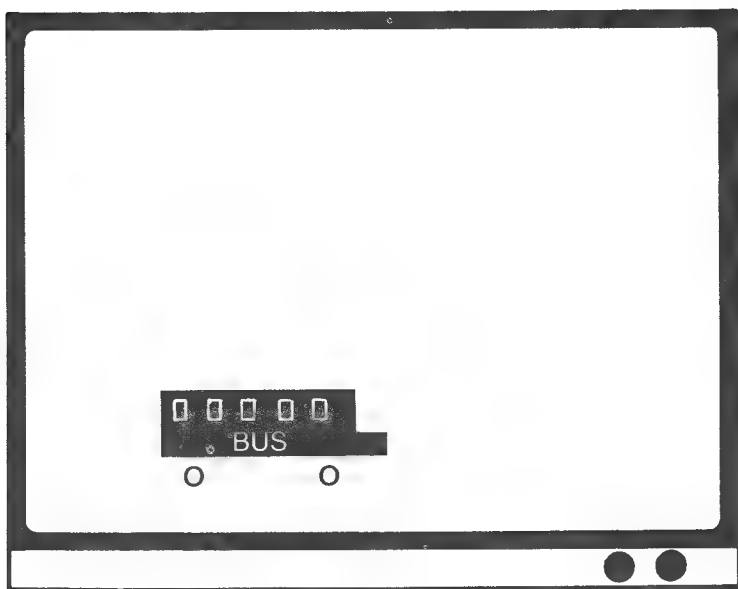
2. Next type space, the letter O, three spaces, and another O.

30 PRINT " O O

2. Make sure that the recorder is properly attached and that the volume level is set.
3. Type LOAD "GAME 2"
4. Press ENTER and the recorder's Play button at the same time.
5. If you need further directions, look back at Lesson 7 in Chapter 1.

## LESSON 5

Having a program like Game 2 that lets you draw is all well and good. But how do you draw things like robots or cars in your own programs? How would you draw this bus, for example?



1. Type the following:

10 PRINT "

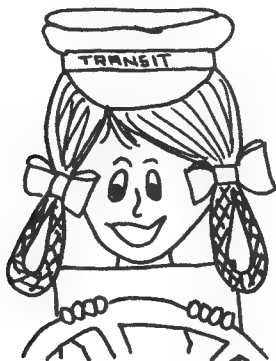
2. Now get Clara set to draw graphics characters by pressing SHIFT and GRAPHICS.
3. Type five of the shape ■ (graphics E) by holding SHIFT and pressing E five times.

10 PRINT " **TTTT**

4. Next it's **█** (graphics 5) so press SHIFT and the 5 key.

10 PRINT " **TTTT█**

5. To type the quotation mark needed at the end of this line we need to get out of graphics mode. Press SHIFT and the GRAPHICS key, then add the quotation mark. ENTER.



6. Type this and then get into graphics mode.

20 PRINT " **BUS**

7. Press the space key, B, U, S, space, space and **█** (hold SHIFT and press 5).

20 PRINT " **BUS**

then press ENTER.

For the last line of our program, we don't need to use graphics at all.

1. Type this:

30 PRINT "

2. Next type space, the letter O, three spaces, and another O.

30 PRINT " O O

3. Add the final quotation mark and ENTER.

Now run the program several times. Can you improve our bus in any way?

### INSECTICIDE PATROL #2



This program has a bug in it. Can you debug it? (Answer on page 127.)

```
10 PRINT "CLARA"  
20 PRINT "HATES"  
30 PRINT "BUGS"
```

## LESSON 6

Your turn again! Now *you* program Clara to draw something.

The next page is a *graphics pad* showing the 'squares' that the screen is ade of. You might find it heplful to draw on the grpahics pad before you begin programming. (There are also more graphics pads at the back of this book.) Here is what the bus looks like drawn on the graphics pad!

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										

Remember that you will need a program line for every row of your picture.

When you have made a good picture, come back to this lesson to learn how to save *your* program on a cassette tape!

\* \* \*

Is your picture program all finished now? Read on to find out how to save a program onto a cassette.

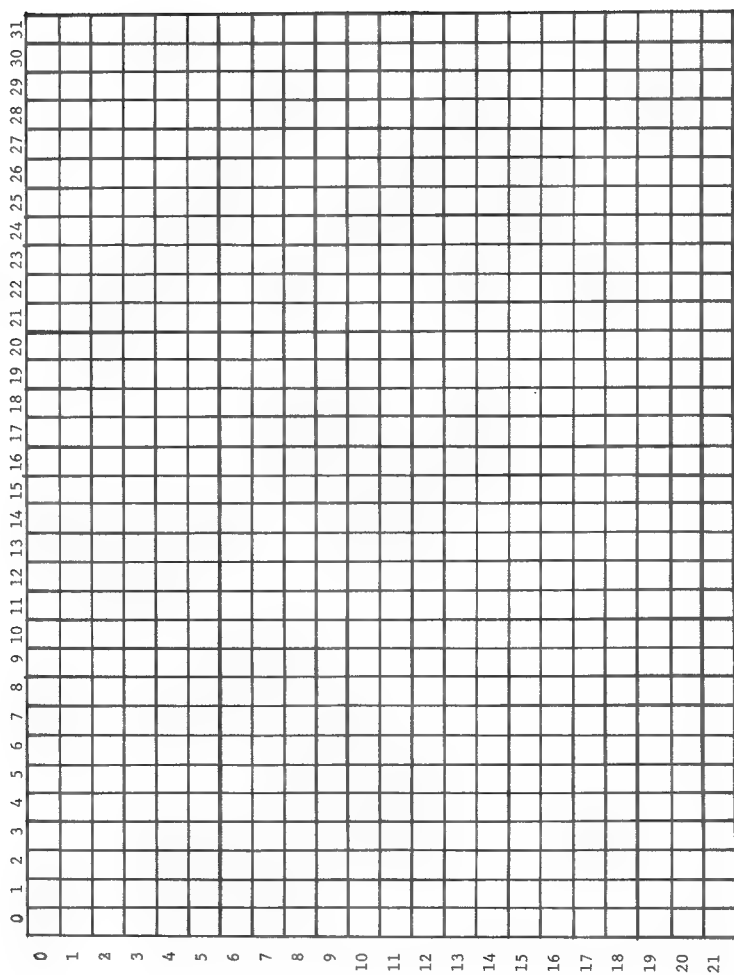
1. Have an adult make sure the tape recorder is properly attached, or look at page 130 for more directions.
2. Put a blank tape of your own (*not* a tape that came with this book) into the tape recorder and rewind it.
3. Press the tape recorder's 'Fast Forward' button while you count 1000, 2000, 3000. Then push 'Stop.' (We do this because you can't record on the very beginning of most tapes.)
4. Type the keyword SAVE and a quotation mark.
5. Decide on a name for your program and then type the name. Remember what the name is, or write the name on the cassette label.

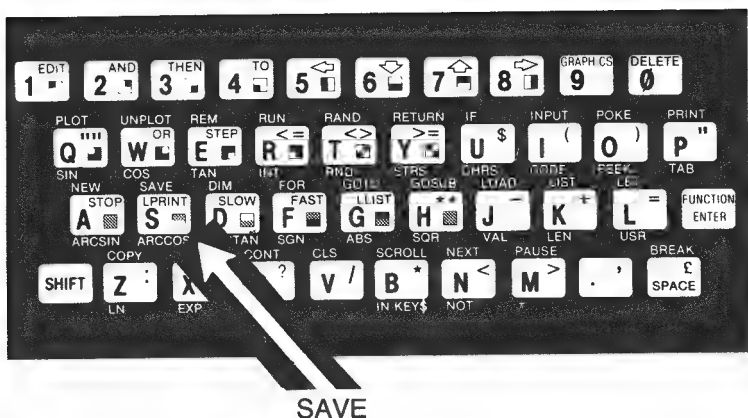
SAVE " program name

6. Type another quotation mark, but don't ENTER yet.

SAVE "program name"







7. On the tape recorder, press Play and Record at the same time.
8. Now press ENTER. The screen will go crazy, just as it does when loading.

When the screen returns to normal, stop the tape recorder. Anytime you want to run your picture program, load it from your tape using the name you just gave it.

**LOAD "program name"**

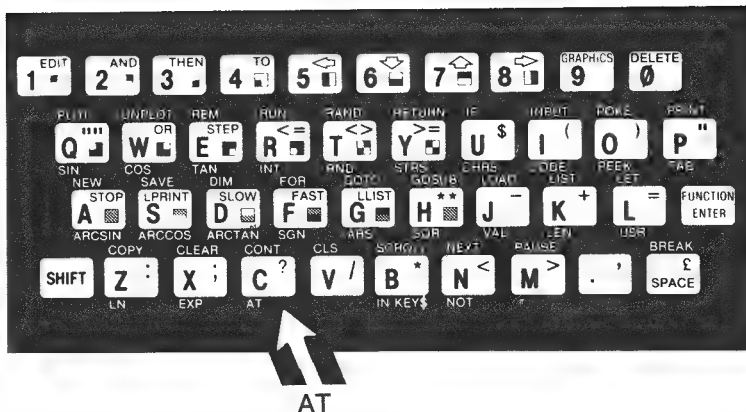
## Review \*\*\*\*\*

**SAVE**—tells the computer to feed the program into the tape recorder. You also need the name of the program, in quotation marks.

## LESSON 7

The screen of a Sinclair computer is divided into 22 rows (going across) and 32 columns (going down). However, the rows aren't numbered from 1 to 22 as you would expect. Instead, the computer numbers them from 0 to 21. The columns are numbered from 0 to 31. Clara doesn't even know how to count like a person! Look at the graphics pad to find the row and column numbers.

These numbers let us print at any part of the screen we want, using a command called PRINT AT. You already know how to type PRINT. But how do we type AT? If you find it on the keyboard, you see it isn't a keyword, or a red character, or a graphics character.



1. Type PRINT.
2. Hold SHIFT while you press FUNCTION, on the ENTER key. Now Clara shows **F**-which means Clara is expecting a *Function*. Functions are simply the words written *below* the keys, like AT.



3. Press the AT key. Clara goes back to expecting a normal letter, **L**, all by herself.
4. Now we need a row number going down the screen (from 0 to 21), comma, and a column number going across the screen (from 0 to 31). For example:

PRINT AT 18, 10

5. Hold SHIFT while you press the X key to type ; (which is

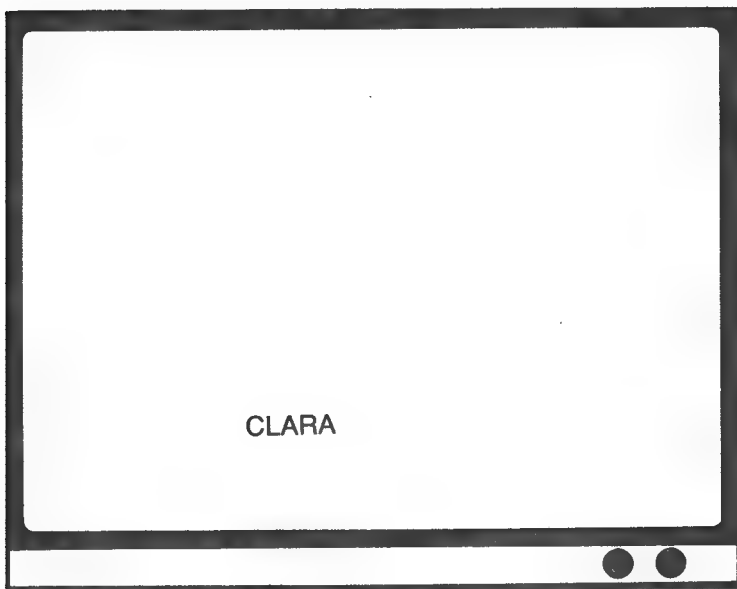
called a *semicolon*). Always remember the semicolon or Clara won't understand you!

```
PRINT AT 18, 10;
```

6. Finish the line like this.

```
PRINT AT 18, 10; "CLARA"
```

When you press ENTER this is what you see.



Try printing your name at several different places on the screen. Be sure your rows don't go higher than 21 or your columns higher than 31—those numbers aren't even on the screen!

**Review** \*\*\*\*\*

1. PRINT AT—this command lets us print anywhere on the screen we wish. You need a row number (0 to 21), a comma, a column number (0 to 31), and then a semicolon (;). For example:

```
PRINT AT 5, 22; "HELP"
```

2. Functions—the words below the keys, such as AT, are function keys. To type functions, first hold SHIFT while you press FUNCTION on the ENTER key. Then press the function key you want.

### INSECTICIDE PATROL #3



This program has *two* bugs in it. Can you debug it? (Answer on page 127.)

```
10 PRINT AT 18,5 "CLARA"  
20 PRINT AT 19, 6; "HATES"  
30 PRINT AT 20,32; "BUGS"  
40 CLS
```

### LESSON 8

We're going to use PRINT AT to make a moving picture. To begin, we'll print a face, toward the middle of the screen. I've used the graphics pad to find where to print the parts of the face.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0																							
1																							
2																							
3																							
4																							
5																*	*	*	*	*			
6														(		0		0		)			
7																	T						
8																-	-	-					
9																							
10																							
11																							
12																							

Type and run this program to draw the face. The spaces in line 20 are very important, so I've shown them by putting periods. Whenever you see a period, type a space.

```

10 PRINT AT 5, 14; "*****"
20 PRINT AT 6, 13; "(.0.0.)"
30 PRINT AT 7, 16; "T"
40 PRINT AT 8, 15; "---"

```

If we want to make this fellow talk we need to open his mouth. Look at the graphics pad to find the row and column numbers where the mouth begins. (Row 8, column 15.) If we print an open mouth at this place, it takes the place of the closed mouth. Add this line to the program. The spaces are needed to erase the corners of the other mouth.

```

50 PRINT AT 8, 15; ".O."

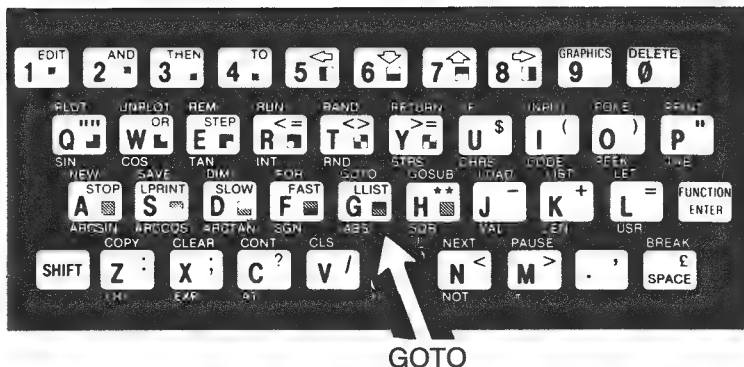
```

If you run the program and look *very* quickly you'll see our fellow opening his mouth. This would be a more impressive program, though, if he would keep opening and closing his mouth as if he were talking. Add this line:

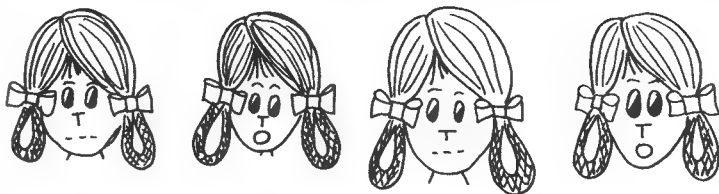
```

60 GOTO 10

```



GOTO is a simple command. All it does is tell Clara to go to line 10 and do the whole program again and again and again.



When you run the program, you'll find two problems.

1. The program *never* stops—it just keeps going back to line 10. To stop it press BREAK, on the space key.
2. The face blinks too quickly. We need the mouth to pause between blinks. Luckily, Clara has a keyword called PAUSE!



We can't just tell the computer to pause, though. It needs to know how long to pause. Add this line to make the face pause between blinks.

45 PAUSE 20

Notice that when you ENTER this line, Clara lists it between line 40 and line 50.

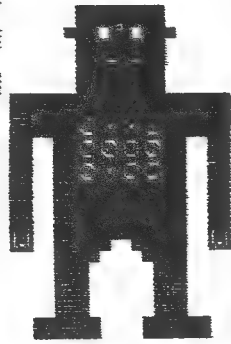
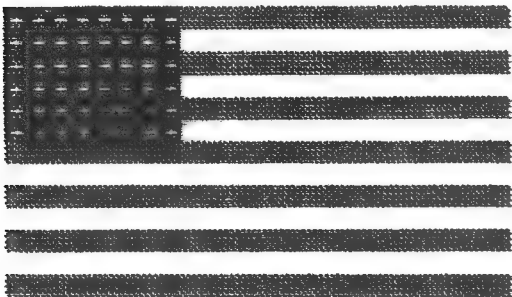
You can, of course, make the pause longer or shorter by retyping line 45 with different numbers. Try running the program with several different pause lengths till you find one you like.

Review \*\*\*\*\*

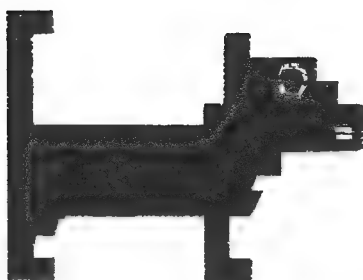
- 1. GOTO—tells the computer to go to another line number (GOTO 10, GOTO 230, etc.).
- 2. BREAK—tells the computer to stop running the program, even if it isn't finished.
- 3. PAUSE—tells the computer to wait before going on to the next program line. You must type a number after PAUSE to show how long to wait.

LESSON 9

Your turn again! Below are several pictures. Can you write programs to make Clara draw any or all of these pictures?







Can you write this program so the dog wags its tail?



#### INSECTICIDE PATROL #4



This program has a bug. Can you debug it? (Answers on page 127.)

```
10 PRINT "CLARA"  
20 PRINT AT 5,5; "HATES"  
30 PRINT AT 10,10; "BUGS"  
40 PAUSE  
50 CLS
```

## LESSON 10

Quiz time again! As before, the questions are below and the answers will appear on the TV screen.

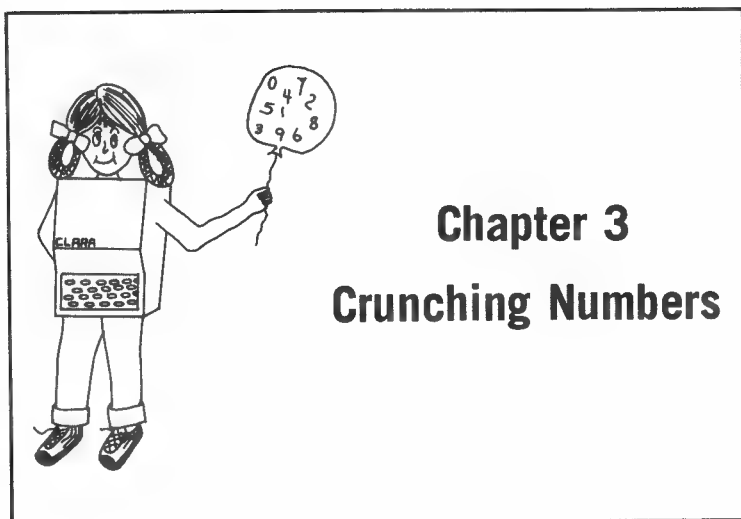
1. Make sure the tape recorder is properly connected to the computer and the volume level is correctly set.
2. Put the Chapter 2 tape into the recorder and rewind it. (See page 136.)
3. Type LOAD "QUIZ 2"
4. Press 'Play' and ENTER.
5. Read question 1 below.
6. Find the best answer from the screen and type the *number* of your choice.
7. Clara will give you two chances before she tells you the correct answer.



## QUIZ 2

1. Which of these is not a function?
2. Clara shows **L**. When you press the R key what will be printed on the screen?
3. Clara shows **G**. When you hold SHIFT while you press the R key, what will be printed on the screen?
4. Clara shows **G**. When you press the R key *without* holding SHIFT, what will be printed on the screen?

5. Which of these is a keyword, or command?
6. Which keyword tells Clara to put the program in her memory onto a cassette tape?
7. Which command tells Clara to wait for a while?
8. To make Clara show **F** you must hold SHIFT while you press which key?
9. Which command makes the screen go blank?
10. Which of these keys do you need to press before you will be able to print any of the graphics characters like **■** ?



## Chapter 3

# Crunching Numbers

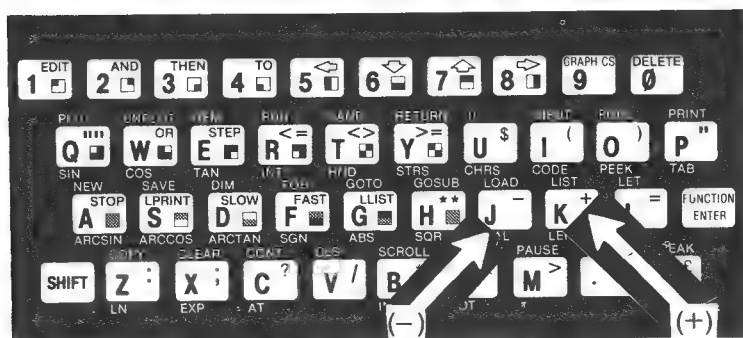
### LESSON 1

When most people think of computers, they think of arithmetic—computers solving long math problems, calculating lunar orbits, keeping track of business costs, and so on. Clara is *very* good at numbers.

Have you ever used a calculator? A calculator is really a kind of computer built to do just one job, whereas Clara can do many different jobs. Computers *can* be used as calculators, though. Here's how.

1. Type PRINT, but don't ENTER yet.
2. Type 2+2, without quotation marks, and ENTER.

PRINT 2+2



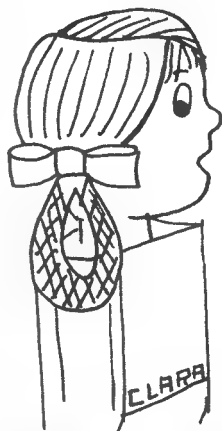
Let's try some more. Type and enter each of these problems, then try a few problems of your own.

PRINT 536-279

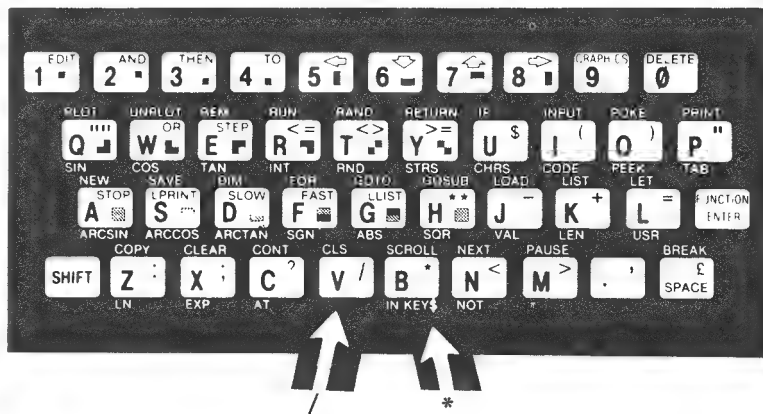
PRINT 8436+1564

Simple, isn't it? But can you figure out how to do multiplication and division?

The difficulty with multiplication and division is that computers don't use the signs ( $\times$  and  $\div$ ) that you're used to. If you type PRINT  $8 \times 9$ , Clara doesn't understand. Dumb computer!



On a computer,  
\* means multiply ( $4 * 8$  is  $4 \times 8$ )  
/ means divide ( $32 / 8$  is  $32 \div 8$ )



What does  $81 / 9$  mean? What about  $76 * 76$ ?  $284 / 4$ ?

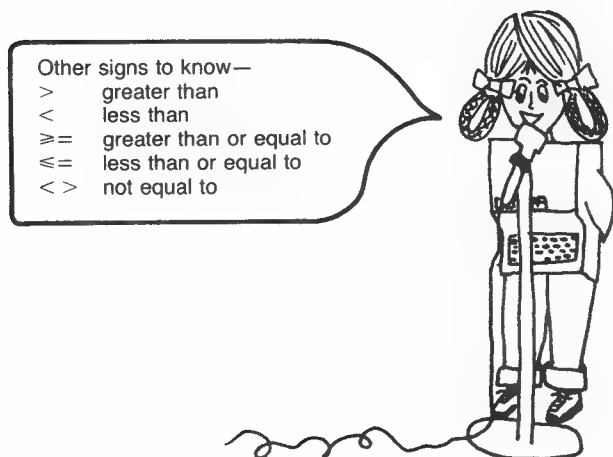
Complicated problems can be typed using brackets, ( ), and the

computer will do the part in the brackets first. For example,  $8 + (7 * 2)$  is  $8 + 14$ , or 22.

For practice, make Clara do these problems and then some of your own. If you feel like it, check Clara by figuring out the answers yourself.

1.  $27 * 174$  (Remember to type *PRINT*  $27 * 174$ )
2.  $8793/56$
3.  $(44 * 44) + 29$
4.  $56 \times 8$
5.  $(46 \times 31) - (128 \div 8)$

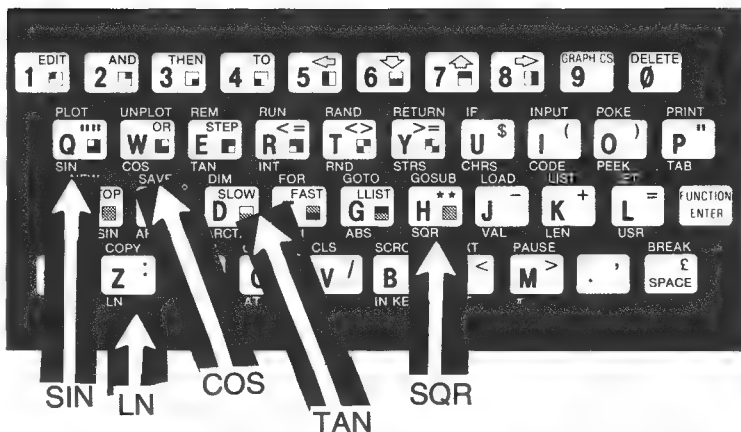
Did you remember to change  $\div$  to  $/$  and  $\times$  to  $*$  in problems 4 and 5?



Like many calculators, Clara can easily figure out such things as logarithms and square roots. For example, to find the square root of 9 type this.

**PRINT SQR 9**

Since SQR is a function, found below the H key, you must first hold SHIFT while you press ENTER to make Clara show **F**. Then press the key above SQR.



Other functions to know—

LN natural logarithm

SIN sine

COS cosine

TAN tangent

Don't worry if your math isn't up to understanding all of these functions.

You will still be able to use the computer as a calculator.

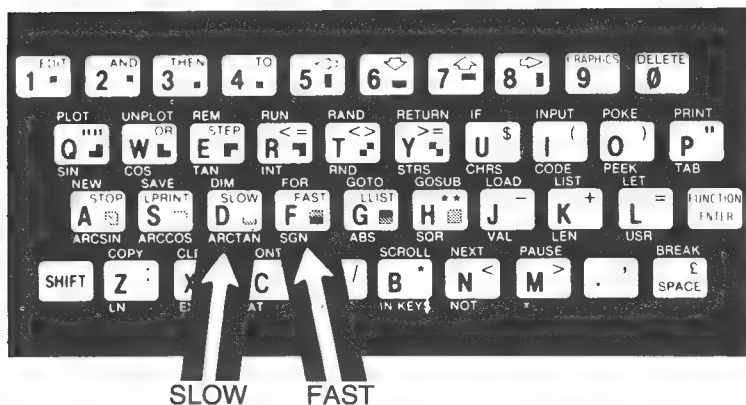
## LESSON 2

Can you do arithmetic quicker than Clara? I don't mean  $2+2$ ; I mean problems like  $8798 + 456789$ . You probably can't. I know *I* sure can't. Yet Sinclair computers are very slow compared to other computers. We *can* speed things up a bit, however.

I can go fast



1. Hold SHIFT while you press FAST.
2. Type several long problems, like `PRINT 846 * 929`, or `PRINT (9 * 32)+(624/12)`.



Notice the way the screen flickers every time you press a key. Clara speeds up by doing strange things to the screen, so we don't usually use FAST while running a program. FAST can be a big help, though, when using the computer as a calculator or typing in a long program.

Use Clara to check the answers to your arithmetic homework. (Notice that I said *check*, not *do*. I don't want to get in trouble with your teacher!) Do some of the problems using FAST and some using SLOW. Which do you like better?



To get the computer back to normal,  
hold SHIFT while you press SLOW.



## Review \*\*\*\*\*

1. FAST—tells the computer to hurry.
2. SLOW—tells the computer to go back to usual speed.

## LESSON 3

Think of a box, covered in Christmas wrapping, with your name on it. Because it is croaking, you know that inside the box are some toads to add to the six toads in your collection. (Doesn't everybody collect toads?) However, you don't know how many are in the box. Maybe two, or three, or 50. The size of your collection, then, can be expressed as  $BOX + 6$ . Any number might be in the box.

Two other boxes are under the tree for you. You know that the boxes both contain ball bearings for you to start a ball bearing collection. Once again, you don't know how many are in each box. Since there are two boxes, we'll keep them straight by calling them box A and box B. The size of your new ball bearing collection, then, looks like this:  $A + B$ . B might be any number. A might be any number.

In this example, the terms BOX, A, and B have a special name. Since they can stand for almost any number, they are called *variables*. Variables can be used in subtraction, multiplication, or divisions as well as addition. Clara uses variables a lot, with names like A, DR, X or M2. Variables are important for a computer, and you'll be learning many, many ways to use them.

How many ball bearings and toads are there? Wait till Christmas to find out!



**Review \*\*\*\*\***

Variable—using a letter to stand for a number.

**LESSON 4**

How does Clara use variables? Let's learn one way by looking at a program. First type NEW. Then type each line of the program.



10 LET A = 8

The keyword LET tells the computer to make a little 'box' in

its memory and to name this 'box' A. Into this 'box,' it puts the number 8. Type the next line.

20 LET B = 9

This tells the computer to make another 'box' named B, and to put the number 9 into this 'box.' Go on to the next program line.

30 PRINT A + B

Can you figure out what this line will do? If we had put quotation marks around it Clara would have printed A+B. But since there are no quotation marks, the computer will act like a calculator and will find the *answer* to A+B.

The computer searches its memory to find variable A, and looks at the 8 in the 'box.' Then it searches its memory for variable B, and looks at the 9 in the 'box.' Now it can do the problem! A + B is the same as 8 + 9! Run the program to see Clara's answer.

Can you write a short program, using the command LET, which will multiply three variables? Try it.

**Review** \*\*\*\*\*

LET—tells the computer to make a 'box' in its memory. LET should be followed by the name of the variable an equals sign and a number (LET D=609). The computer puts 609 into the 'box' named D.

## LESSON 5

Try this problem on the computer—9 million times 9 million.

PRINT 9000000 \* 9000000

What's the answer—8.1E+13???? What does *that* mean?

When a number is really large, like the answer to our problem (81 trillion), the computer doesn't like to print all those zeroes. Instead, Clara tells you how many *places* there are up to the decimal point, and drops all the zeroes.

The E+ tells us that the computer isn't printing the whole answer, just showing the number of places (ones, tens, hundreds, thousands, ten thousands, etc.). Thus 8.1E+13 really means 81,000,000,000,000 (13 places).



For another example,  $5.37E+15$  is 5,370,000,000,000,000 (15 places).

Make up some problems using very big numbers. Are any of the answers printed using E+?

\* \* \*

Time for another game!

In Game 3, Clara hides a treasure on the screen. Your job is to guess the row and column numbers of the buried treasure. Find that treasure!

1. Make sure the tape recorder is properly connected, and that the volume level is correctly adjusted.
2. Put the Chapter 3 tape into the recorder and rewind it. (See page 137.)
3. Type LOAD "GAME 3"
4. Press ENTER and Play.

## LESSON 6

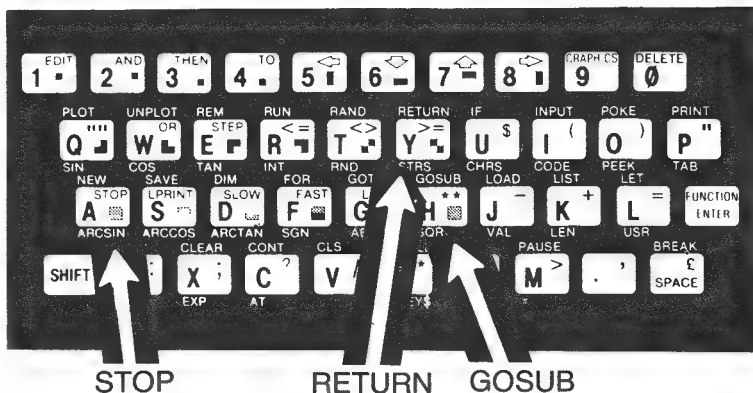
How much would you weigh on the Moon? On Mars? On Jupiter? We are going to write a program which will tell you the answers.

It uses a new command, GOSUB, which means GO to a SUB-routine. A subroutine is a small program that a larger program uses over and over. To save time and space a subroutine can be typed just once, usually at the end of a program.

```
10 LET WEIGHT = (however much you weigh)
```

Notice that we can use whole words to name variables.

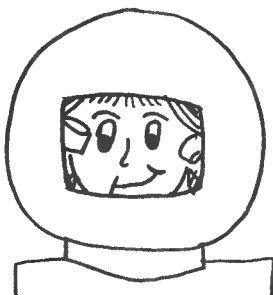
```
20 PRINT "MOON WEIGHT"
30 LET GRAVITY = .16
40 GOSUB 200
```



```
50 PRINT "JUPITER WEIGHT"
60 LET GRAVITY = 2.6
70 GOSUB 200
190 STOP
```

Line 190 is needed to stop Clara from 'crashing' into the subroutine when the program is over. Here is the subroutine itself, which calculates your weight on the different planets.

```
200 LET W = WEIGHT * GRAVITY
210 PRINT W
220 RETURN
```



When the subroutine is finished, RETURN tells the computer to return to where it came from in the main program.

Can you add more lines to the main program to make it print your weight on still more planets? The GRAVITY on other planets is shown below.

Mercury	.37
Venus	.9
Earth	1.0
Mars	.38
Saturn	1.16
Uranus	.96
Neptune	1.43

**Review \*\*\*\*\***

- 1. GOSUB—tells the computer to go to a subroutine. You must include the line number of the beginning of the subroutine (GOSUB 1000).
- 2. RETURN—tells the computer to return to the main program. Every GOSUB must have a RETURN.
- 3. STOP—tells the computer to end the program.

**INSECTICIDE PATROL #5**



This program has a bug. Can you debug it? (Answer on page 128.)

```
10 PRINT "CLARA"  
20 GOSUB 70  
30 PRINT "HATES"  
40 GOSUB 70  
50 PRINT "BUGS"  
60 STOP  
70 PRINT  
80 PRINT
```

## LESSON 7

Do you know much about the metric system? The program in this lesson uses variables to change from miles to kilometers.

```
10 LET M=75  
20 LET KM=M*1.61  
30 PRINT "MILES"  
40 PRINT M  
50 PRINT "KILOMETERS"  
60 PRINT KM
```

Rewrite line 10 put a different number of miles into the variable M.

This program works, but we can only figure out one distance each time we run the program. Let's improve the program so it will do several numbers. We'll turn lines 20 to 60 into a subroutine, to be used over and over. We need to add a RETURN line.

```
70 RETURN
```

We will now add lines at the beginning of the program to let us figure out several distances.

```
2 LET M=999  
4 GOSUB 20  
6 LET M=375  
8 GOSUB 20  
10 LET M=65  
12 GOSUB 20
```

```
14 LET M=19832
16 GOSUB 20
18 STOP
```

Notice that lines 6, 10, and 14 keep changing the number in the variable box M. Variables often change in the middle of a program.

Now, put numbers of your own into lines 6, 10 and 14. How many kilometers is it to school? How many kilometers to the next town?

## LESSON 8

Computers use variables in very many ways. Here are a few more examples.

Variables can be used almost anywhere a number can be used. This includes PAUSE, GOTO, GOSUB and PRINT AT. Type the following program, which shows how this is done.

```
10 LET X=14
20 LET Y=16
30 PRINT AT X,Y; "GOOD"
40 PAUSE X*Y
50 PRINT AT Y, X; "BYE"
60 PAUSE X*Y
70 CLS
80 GOTO X+Y
```



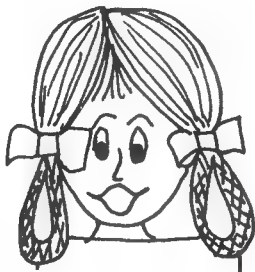
This is another program that never stops on its own. Do you remember how to stop it? Press BREAK.



In lines 30 and 50, variables take the place of the row and column numbers. In lines 40 and 60, the product of the two variables tells Clara to PAUSE 224 (14 \* 16 is 224). Line 80 adds up 14 +16 and so the computer goes to line 30.

Variables can also be used to count. Try this program.

Do you remember to type NEW  
before starting a new program?



```
10 LET C=1
20 PRINT C
30 LET C=C+1
40 GOTO 20
```

Line 30 is an interesting one! It tells the computer to add 1 to the variable C, making C equal to 2. The computer then jumps back to line 20 and prints C (which is 2), then goes to line 30 and makes C=3, jumps back to line 20 and prints 3, makes C=4, and on and on. It just keeps going till it fills the screen!

To count by twos (2, 4, 6, 8), we would make these changes.

```
10 LET C=2
30 LET C=C+2
```

Can you make this program count by fives? Can you make this program count by a weird number, like 127?

## LESSON 9

Your turn once more! Your assignment is to write a program that uses as many of the BASIC words you've learned as possible. You decide what the program will do. To refresh your memory, I've listed all the commands we've covered so far.

PRINT  
CLS

PAUSE  
NEW

GRAPHICS  
SLOW  
GOSUB  
LET  
RUN  
PRINT (without quotation marks)  
SAVE

FAST  
PRINT AT  
RETURN  
STOP  
LOAD

If you think your program is good enough, save it on a cassette tape.

Happy programming!

## LESSON 10

Quiz time again!

After you have set up the tape recorder and rewound the Chapter 3 tape, type LOAD "QUIZ 3". (See page 139.) Press Play and ENTER.

As before, the questions are in the book, while the answers appear on the TV screen. Do you think you'll get them all right?



## QUIZ 3

1. Which sign means 'times' or 'multiply'?
2. Which sign means 'divided by'?
3. Which sign means 'greater than'?
4. Which sign means 'less than or equal to'?
5. Which keyword tells the computer to make a 'box' in its memory for a variable?
6. One of these commands changes the computer's speed, but sort of messes up the TV picture. Which one?
7. Which command tells Clara to go to a subroutine?
8. Every program which uses GOSUB must use another keyword

at the end of the main program. Which command is this?

The rest of the questions are about this program. Study it, then go on to questions 9 to 12.

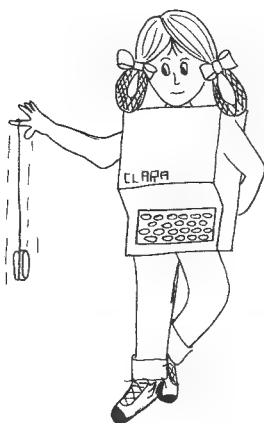
```
10 LET A=10
20 LET R=4
30 GOSUB 80
40 LET T=R*(C/1)
50 PRINT T
60 STOP
70 LET R=2
80 LET C=A/A
90 LET C=C+1
100 RETURN
```

9. How many different variables are used?
10. What program line will Clara do right after the RETURN in line 100?
11. Clara will never get to one certain line of this program. Which line is it?
12. What number will Clara print on the TV when you run this program?

# Chapter 4

## Here We Go

### Loop the Loop



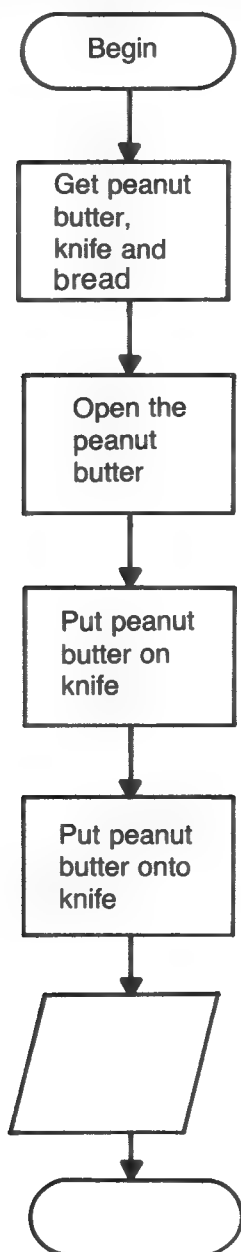
#### LESSON 1

Think of a delicious peanut butter sandwich. Here are the steps needed to make that peanut butter sandwich:

1. Gather peanut butter, knife, and two slices of bread.
2. Open the peanut butter jar.
3. Scoop some peanut butter onto the knife.
4. Spread the peanut butter onto one slice of bread.
5. Put the two slices of bread together.
6. Eat!

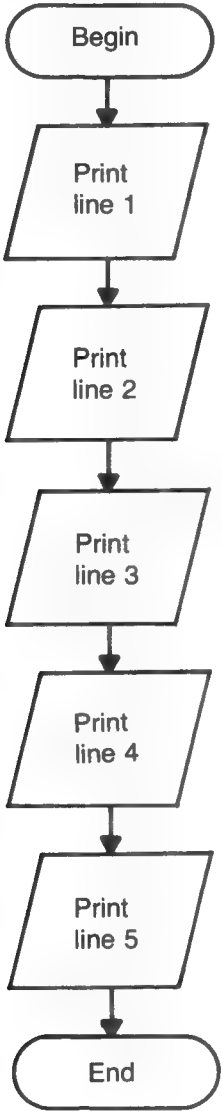


We can draw a special kind of picture of these directions, called a flowchart. Below is the peanut butter flowchart.



The last two parts of the flowchart are blank. Do you know what should go in them?

We can also draw flowcharts of computer programs. They can help make a program easier to understand. As an example, let's look at a flowchart of a program to cover five lines of the screen with the letter X.



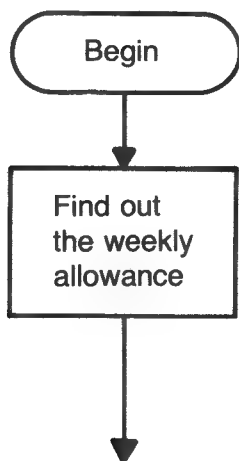
This same program, written in BASIC, is as follows.

```
10 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
20 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
30 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
40 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
50 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

Can you see how each step in this program is shown in the flowchart?

Below is a short program that will calculate how much allowance you get in a year. Can you finish the flowchart of this program? Try it.

```
10 LET ALLOWANCE=1.25 (Or whatever your
                        weekly allowance is,
                        in cents. Leave out
                        the dollar sign!)
20 LET YEAR=ALLOWANCE * 52
30 PRINT "MY YEARLY ALLOWANCE IS"
40 PRINT YEAR
```



Can you finish this flowchart?

**Review** \*\*\*\*\*

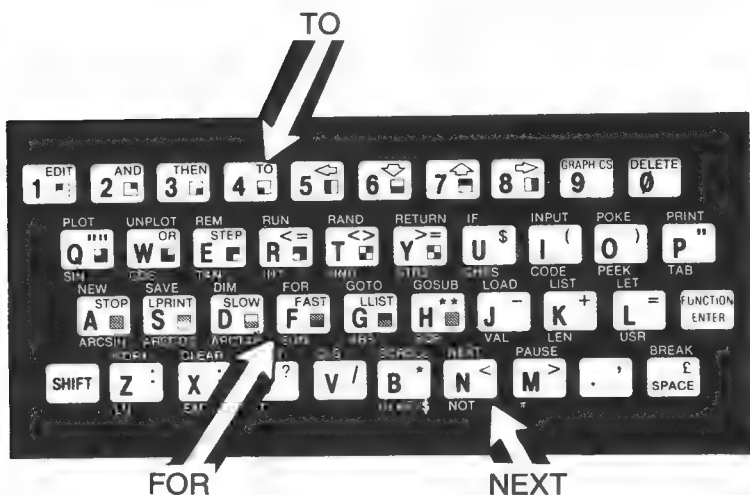
Flowchart—a 'picture' of a computer program that shows how the program works.

## LESSON 2

Let's take a closer look at the program from Lesson 1 that printed a bunch of X's on the screen. If you typed this program, you realize that it was a lot of work. Computers are supposed to be great at doing things over and over. Isn't there an easier way to print all those X's?

Would you believe we can rewrite this whole program into just three short program lines? Type this program.

```
10 FOR T=1 TO 160
20 PRINT "X";
30 NEXT T
```



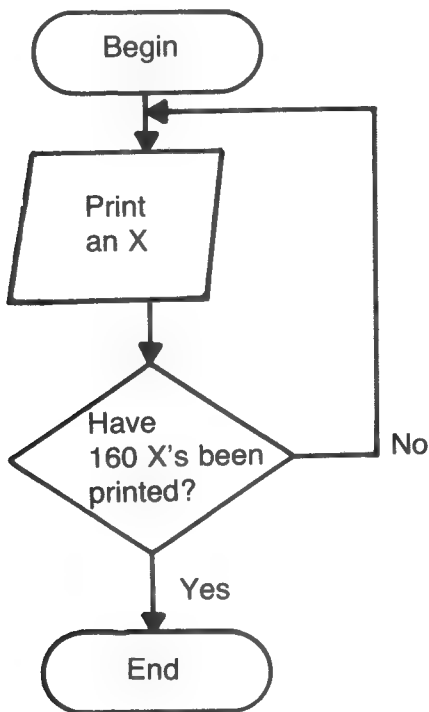
Lines 10 and 30 of the program go together. Every FOR....TO.. line must have a NEXT line. These keywords tell Clara to keep going from line 30 back to line 10 one hundred and sixty times. When the computer has finished 160 loops, it goes on to the next program line, if there is one.

```
10 FOR T=1 to 160
20 PRINT "X"           160 times
30 NEXT T
```

Can you figure out what the semicolon (;) in line 20 did? It made the X's print right next to each other. Can you rewrite line 20 to print something other than X's?



The flowchart of this program is now quite different, since the program now loops back to its beginning over and over. Can you follow the arrows now?



FOR . . . NEXT loops, as these are called, have many uses in programming. For example, a loop can be used to make flashing words. Type NEW, then enter this program.

```
10 FOR I=1 TO 500
20 PRINT AT 11,11;"LIGHTNING"
30 CLS
40 NEXT I
```

Loops save a lot of typing. Without the loop, the lighting program would start like this.

```
10 PRINT AT 11,11;"LIGHTNING"
20 CLS
30 PRINT AT 11,11;"LIGHTNING"
40 CLS
```



The program would end at line 1010! Yes, loops can save a *lot* of typing!

**Review** \*\*\*\*\*

1. FOR...TO..—tells the computer how many times to loop.  
Examples:

```
FOR X=1 to 3
FOR V=1 TO 200
```

2. NEXT—tells the computer to go back to the beginning of the loop over and over. You must start every loop with a FOR...TO.. line and end the loop with a NEXT.. line.

### LESSON 3

We can teach Clara to count using a FOR....NEXT loop. Type this:

10 FOR J=1 TO 100

Line 10 tells Clara that the loop is to be done 100 times.

20 SCROLL



SCROLL

SCROLL tells Clara to print at the bottom of the screen, and to 'roll' the display upward. Don't worry if you don't understand what I mean right now; you'll see when you run the program. Finish with these lines.

30 PRINT J  
40 NEXT J

When you run this program you can actually see Clara counting through the one hundred loops as the numbers scroll up the screen.

You'll need to use BREAK to stop the program.



We don't have to start at 1 when we count. Let's make Clara count from 1000 to 1100. We can do this by changing just line 10 of our program.

```
10 FOR J=1000 TO 1100
```

Run the program.

Type NEW. Now write a program to make Clara count from 99950 to 1000000. Use SCROLL. Then try to flash the number 1000000 on the screen when the computer reaches the end.

**Review** \*\*\*\*\*

SCROLL—tells the computer to print at the bottom of the screen and 'roll' on up the screen.

## INSECTICIDE PATROL #6



This program has two bugs. Can you debug it? (Answer on page 128.)

```
10 FOR T=1 TO 25
20 PRINT AT T, 2;"CLARA HATES BUGS"
30 CLS
```

## LESSON 4

Now for a computer program about *you*. You'll need to think of some things about yourself to tell Clara—your favorite food, best friend, things you like to do, etc. Then write a fancy program about yourself like the example program in this lesson about Scott. Type in at least part of the example, to learn how FOR...NEXT loops are used to make a flashy display.

```
10 PRINT "MY NAME IS"  
20 FOR F=1 TO 50  
30 PRINT AT 1,2; "          "           (5 spaces)  
40 PRINT AT 1,2;"SCOTT"  
50 NEXT F
```

Do you understand what this loop does? First it prints as many blank spaces as there are letters in the name. Then it prints the name. On the next loop the blank spaces erase the name, but line 40 prints it again. A flashing name!

The rest of the example program also uses loops in the same way. Notice that the same variable, F, is used in each loop

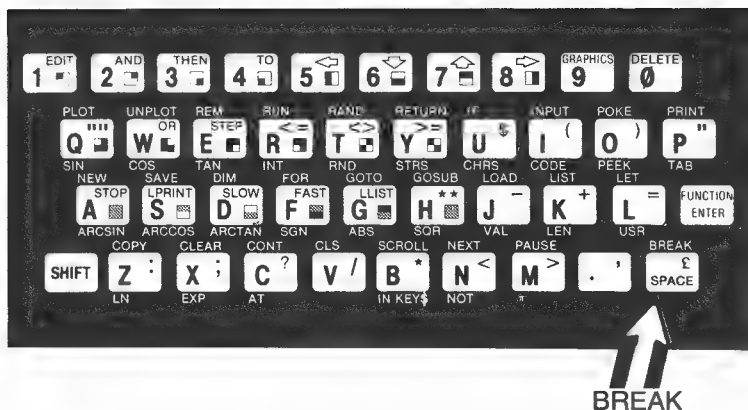
```
60 PRINT "MY FAVORITE TV SHOW IS"  
70 FOR F=1 TO 50  
80 PRINT AT 3,2;"          "           (21 spaces)  "  
90 PRINT AT 3,2;"KIDS SUPER POWER HOUR"  
100 NEXT F
```

```
110 PRINT "MY BIRTHDAY IS"  
120 FOR F=1 TO 50  
130 PRINT AT 5,2;"          "           (11 spaces)  
140 PRINT AT 5,2;"NOVEMBER 23"  
150 NEXT F
```

For a finishing touch, I'll add two lines which will run the program over and over. You'll need to use BREAK to stop it.

```
160 CLS  
170 GOTO 10
```

Have you figured out what you'll put in your program? Try to make it fit the pattern of the example—print the beginning of a sentence, then use a FOR...NEXT loop to flash the end of the



**BREAK**

sentence. Make your program as fancy as you wish. Use graphics characters, inverse printing, the graphics pad, or anything else that improves your work.

## LESSON 5

Are you ready FOR the NEXT game? It is a good example of how FOR...NEXT loops can be used, so after you've tried the game we will study the program.

1. After correctly attaching the tape recorder and setting the volume level, put in the Chapter 4 tape. (See page 140.)
2. Type LOAD "GAME 4"
3. Press Play and ENTER at the same time.

## Game Directions

Your task in this game is to build the Empire State Building, floor by floor!

The game starts after you have pressed the "1" key as directed. A BASIC word will appear on the screen. You must find the word on the keyboard and press that key. In this game, you don't need to use SHIFT for the red characters.

Every time you press the correct key, one floor of the Empire State Building will go up. If you take too long, however, Clara will just go on to another BASIC word.

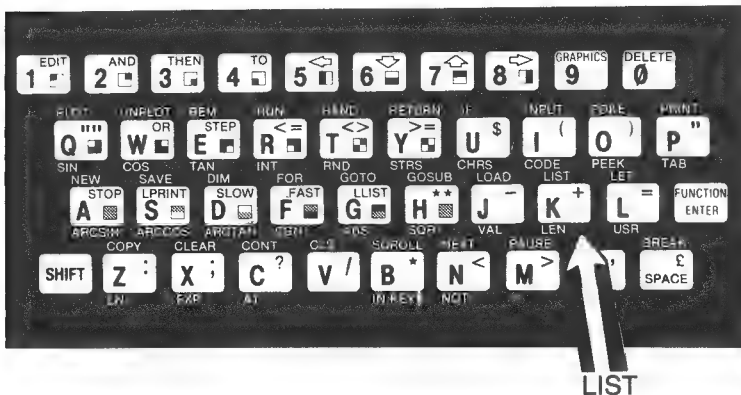
If you manage to build the whole skyscraper in 25 tries, there is a special surprise waiting for you. I won't tell you what the surprise is—but what would you expect if a kid built a real skyscraper single-handedly?

Try the game now to see what it does. Then leave the program

in Clara's memory and come back to this page to learn more about *how* it works.

\* \* \*

Are you ready to study the program now? Good. Let's start at the very beginning. Type LIST.



This command tells Clara to show the program lines, starting at the beginning.

Many of the program lines won't make much sense to you, but lines 70 and 80 seem clear. Line 70 sets up a loop and line 80 finishes the loop—with nothing in between! Clara will just go back to line 70 one hundred times—and that takes a second or so. Meanwhile, the title, GAME 4, is being shown on the screen. This loop was just used to make Clara pause. After the loop, the screen is cleared.


Line 120 also starts a FOR...NEXT loop. This line gives you 25 tries to build the skyscraper.

If you want more tries or fewer tries, rewrite line 120 with a different number.



You might have noticed that the whole program will not fit on the screen. To see beyond line 140, type LIST 150. This tells Clara to show the program, starting at line 150. LIST 300 would start a line 300, and so on.

Can you find the beginning of the next loop? Line 180 sets up another FOR...NEXT loop, which determines how long Clara will wait for you to press the correct key.



If you want more time or less time,  
rewrite line 180 with a different  
number.

So far we have found three FOR...NEXT loops. Use the command LIST to look at the rest of the program. How many loops are there altogether? Can you find a loop that counts backwards? Which other loop is used just as a pause? How many of the BASIC commands in this program do you know?

## Review \*\*\*\*\*

LIST—tells the computer to show the program lines, from the very beginning of the program.

LIST 180—tells the computer to show the program, starting with line 180.

## LESSON 6

What happens to Clara when the screen gets filled up? Type this program and let's see.

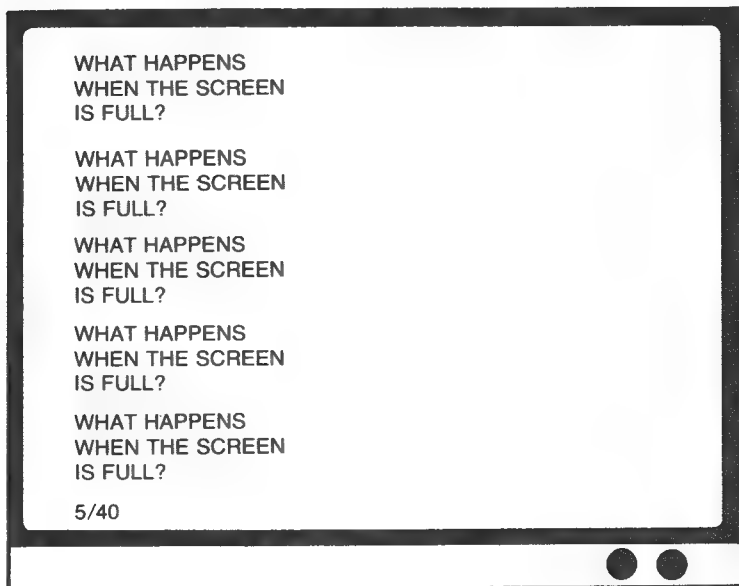
```
10 FOR J=1 TO 10
20 PRINT "WHAT HAPPENS"
30 PRINT "WHEN THE SCREEN"
40 PRINT "IS FULL?"
```



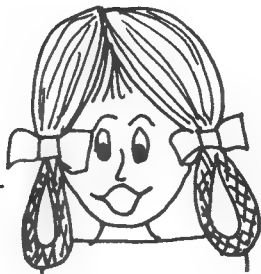
```
50 PRINT  
60 NEXT J
```

(This just prints a blank line)

When you run the program, notice that the loop does not repeat the full 10 times—a bit over five times, actually. Instead, the screen looks like this.



The number 5 at the bottom left of the screen is my way of telling you, "Hey, human, the screen is full. I can't print any more!"



To continue the program, type `CONT` (short for `CONTInue`).

The computer clears the screen, then continues the program right from where it left off!

Rewrite line 10 as follows, then run the program again. How many times must you use `CONTInue` now?



```

10  FOR I=1 TO 64
20  PRINT "■ ■ ■ ■ ■";
30  PRINT "■ ■ ■ ■ ■";
40  NEXT I

```

The difference, of course, is the semicolon at the end of each line. This mark tells Clara not to start a new row when done printing the program line. Run the program to see what our design looks like now.

Wow! Quite a change! That semicolon sure can be used to make neat patterns.

Here is another design program for you to enter. To type line 30, get into graphics mode and type four spaces. To type line 40, get into graphics mode and type five graphics H's.

```

10  FOR I=1 TO 50
20  PRINT "* A *";
30  PRINT "    ";
40  PRINT "HHHHH";
50  NEXT I

```

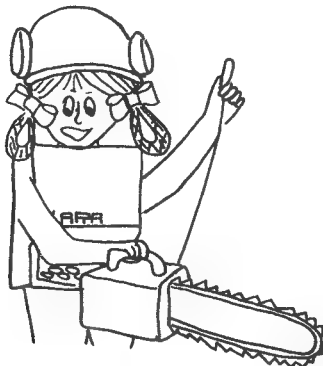
Can you now write a design program of your own? Use a FOR...NEXT loop. Print letters, numbers, inverse letters, graphics characters, anything you want. Just be sure to use the semicolon at the end of each PRINT line. Experiment!

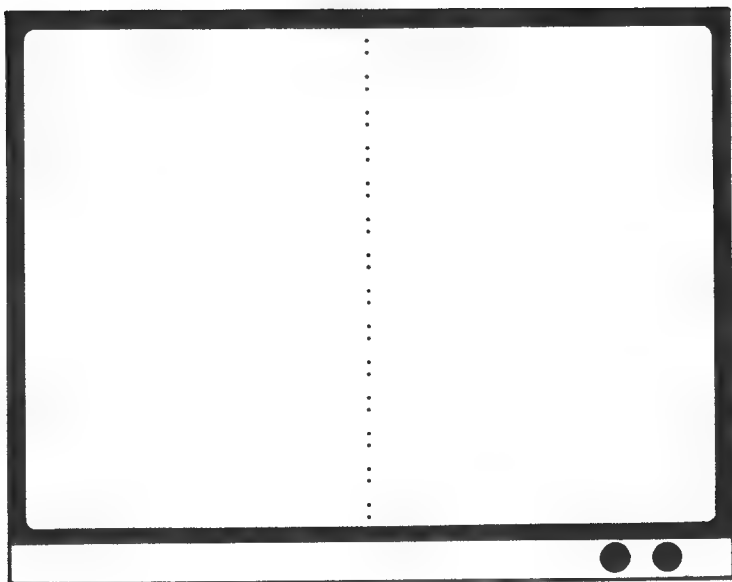
**Review** \*\*\*\*\*

Semicolon—tells the computer not to skip any spaces before the next PRINT line.

## LESSON 8

Imagine cutting the TV screen in half, top to bottom. (Just imagine—*don't* get out a chain saw and try it!)

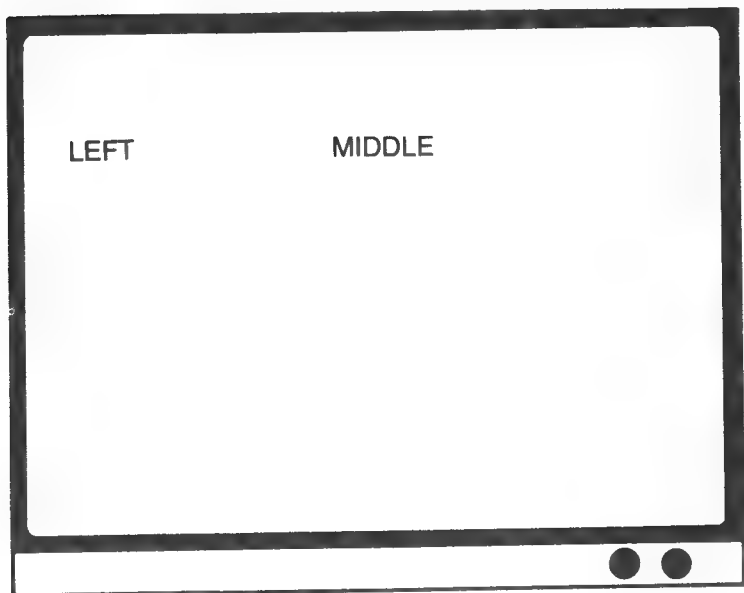




Clara can easily print at our imaginary line, using only a simple comma! The following programs show how this works.

Program 1

```
10 PRINT "LEFT",  
20 PRINT "MIDDLE"
```



The comma tells the computer to jump to the next half of the screen. We can put the comma in several different spots in our program without changing the screen display. Each of these programs gives the same result, even though the comma is moved around.

Program 2

```
10 PRINT "LEFT"  
20 PRINT, "LEFT"
```

Program 3

```
10 PRINT "LEFT", "MIDDLE"
```

The following program *won't* work, however. The comma must be *outside* the quotation marks if you want to jump to the next half of the screen.

```
10 PRINT "LEFT,"  
20 PRINT "MIDDLE"
```

The next program may surprise you a bit when you run it.

```
10 PRINT "THIS LINE GOES PAST THE MIDDLE,"  
20 PRINT "SO WHERE WILL THE COMMA SEND US?"
```

Remember, the comma tells Clara to jump to the *next* half of the screen. Since line 10 has already gone past the middle of the screen, the *next* half is the beginning of the next row. It may help you to remember that  $\frac{1}{2} + \frac{1}{2}$  is 1.

Do you understand the difference between printing with commas and printing with semicolons?



## Review \*\*\*\*\*

Comma—when it is used *outside* the quotation marks of a print line, a comma tells the computer to jump to the next half of the screen before printing anything else.

### INSECTICIDE PATROL #7



This program has two bugs. Can you debug it? (Answer on page 128.)

```
10 PRINT "CLARA ";  
20 PRINT "HATES",  
30 GOTO 10  
40 PRINT "BUGS"  
50 PAUSE B  
60 CLS
```

### LESSON 9

Your turn once more! Use what you've learned in this chapter to help you write a program to count to 200.

It's easy enough to make Clara count, of course. However, can you use **SCROLL** to make the screen look like the following (including all the words)?

THIS PROGRAM COUNTS TO 200.  
IT IS TRICKIER THAN IT SEEMS.  
(BE CAREFUL HOW YOU **SCROLL.**)

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22

Remember that you will need to **SCROLL** after every line that you print.

Good luck!

## LESSON 10

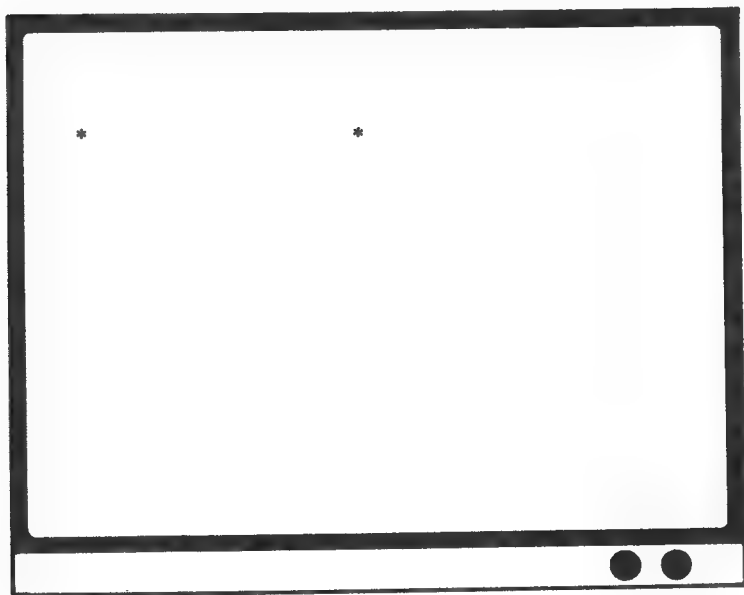
Once more, a computer quiz! As before, you must load the program from the cassette for this chapter (Chapter 4). The name of the program to load is "Quiz 4". (See page 141.)

The questions are in this book and the possible answers are on the computer.



## QUIZ 4

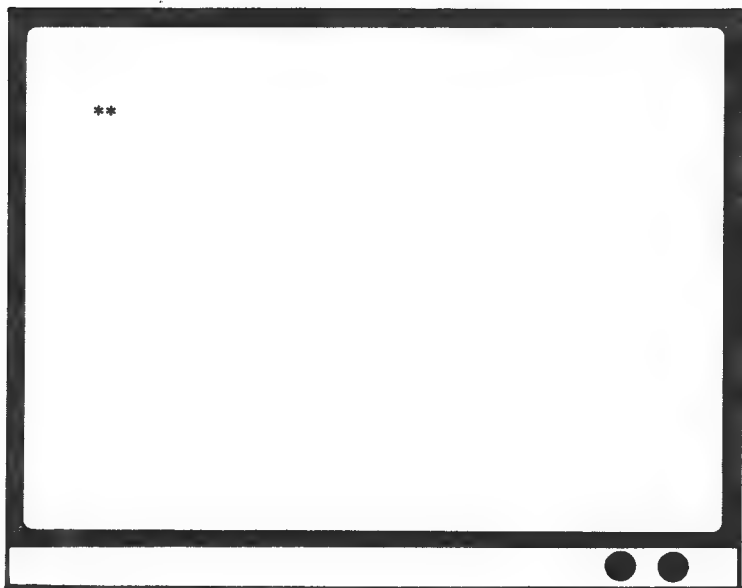
1. This command tells the computer to show the program in its memory, starting with the very beginning.
2. When the screen is full, this command clears the screen and continues the program.
3. This program line will result in a blank line being printed.
4. If you type this, the computer will show you the program in its memory, beginning at line 150.
5. This program line will give the following display.

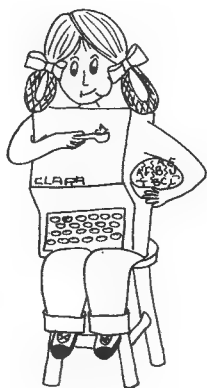


6. Typing this line will give the display shown on the next page.
7. Which of these lines has a bug in it?
8. This keyword tells the computer to print at the bottom of the screen and roll the display upwards.
9. This command tells the computer how many times to loop.
10. You need to have this BASIC word in the *same* line as the command FOR.
11. This command marks the end of a loop.
12. Study this program. What will Clara print?

```
10 LET A=200
20 LET B=35
30 PRINT "A + B"
```







## Chapter 5

### Putting in Input

#### LESSON 1

In the quiz programs and the game programs, the computer asks humans questions. But then, it is easy to make Clara ask a question.

PRINT "HOW OLD ARE YOU?"

The thing is, how do you get Clara to listen to your answer?

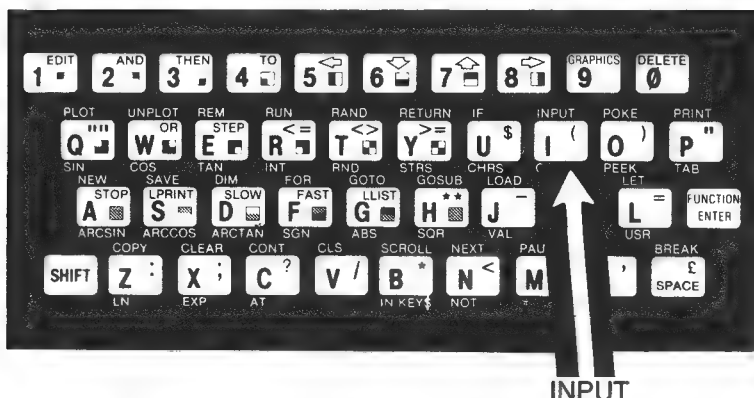


The command INPUT tells the computer to pay attention to the answers you type in. For example, enter this program.

```

10 PRINT "HOW OLD ARE YOU?"
20 INPUT A

```



Notice that we put a variable, A, right after the keyword INPUT. This tells the computer to make a 'box' in its memory labelled A. Clara will put the number that you type into this box. You must *always* put a variable after INPUT, or Clara won't understand you.

Let's write another program that will let you put in two numbers for the computer to multiply.

```

10 PRINT "TYPE THE FIRST NUMBER."
20 INPUT A
30 PRINT "TYPE THE SECOND NUMBER."
40 INPUT B
50 LET C=A*B
60 PRINT A; " X ";B;" = ";C

```

Line 60 is interesting in the way it prints out the equation, complete with 'X' and '='.

Run this program several times. Can you input such big numbers that Clara drops the zeroes and uses E+ in the answer?

## Review \*\*\*\*\*

INPUT—tells the computer to wait for you to put in data while a program is running.

## LESSON 2

In Chapter 3 you wrote a program to check your math

homework. Now, using INPUT, we can write a really short program to find the answers to *any* kind of math problem. Type this program.

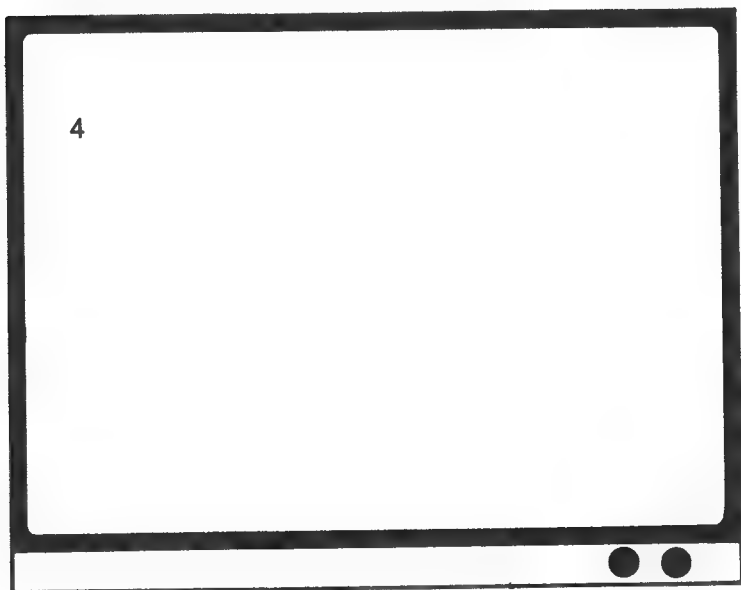
```
10 INPUT Y
20 PRINT Y
```

That's all there is to the program! However, when you run it, nothing seems to happen. Actually, though, Clara is patiently waiting for you to type your input. Notice the bottom left hand corner of the screen. Clara is showing an **L** rather than a **K**, as you might expect. The **L** shows that Clara is waiting for input.

If you type the number 15 and then enter, the computer prints 15 also. However, line 10 will also let us put an *equation* into the computer. Run the program, then carefully type this equation:

$$2 + 2$$

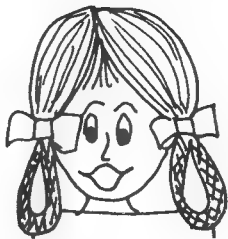
Presto! Clara prints not the *equation*, but the *answer*!



This happens because Clara doesn't keep the whole equation in the variable box Y. Instead, the computer figures out the answer to the equation and puts the *answer* into box Y. So, when line 20 tells Clara to PRINT Y, the answer comes out. This also works for division, multiplication, subtraction and functions such as SQR and SIN.

Our short program works, but it would be kind of awkward to use to check a whole homework assignment. It is always a good idea to make your programs as easy to use as possible, so let's improve the program with these lines. You can see how handy it is to number program lines by 10s, so that new lines can be added like this.

Programs that are easy to use are called 'user friendly.' They are also the most fun to use!



```
1 PRINT "HOW MANY PROBLEMS?"  
2 INPUT X  
3 PRINT
```

Line 1 tells the user of the program to type the number of problems he wants to do. Line 2 stores this data in variable X, while line 3 just tells Clara to skip a line.

```
4 FOR P=1 TO X  
5 PRINT "PROBLEM ";P,
```

Line 4 sets up a loop that will run once for each problem. Line 5 prints the problem, P, that we are on. The comma at the end of the line makes the answer, which is printed by line 20, appear halfway across the screen.

Finish the program by ending the loop.

```
30 NEXT P
```

Happy homework checking!

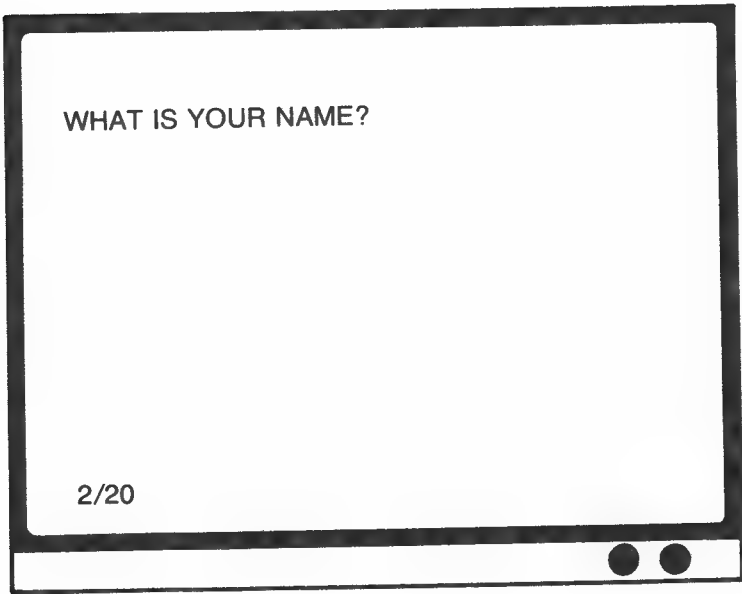


### LESSON 3

Can you make Clara ask you your name? Will this work? Try it.

```
10 PRINT "WHAT IS YOUR NAME?"
20 INPUT N
30 PRINT N
```

Well, *that* doesn't work.



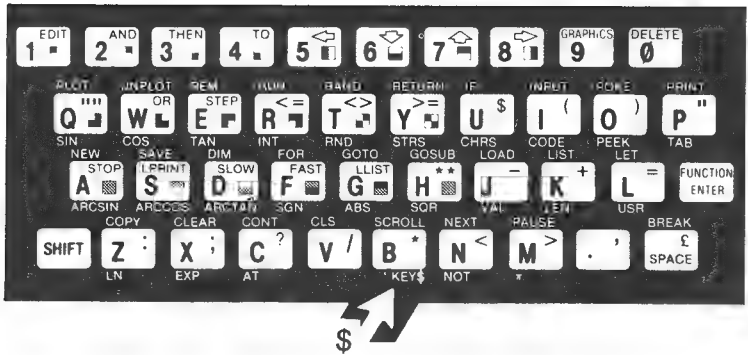
We need a different kind of variable here, because boxes like AB, Y, or N can only hold numbers, not letters or words.

Clara uses *string variables* to hold letters or words. 'String' doesn't mean the variable box is wrapped in string, however. It is just a name for this kind of variable. Unlike number variables, string variable names can only be one letter long.

To change variable N to a string variable, we need only add a dollar sign, making it N\$—which we now pronounce "N string."

Change lines 20 and 30 of our program to the following, then run it.

```
20 INPUT N$
30 PRINT N$
```



That's better! We can also make string variable boxes using the command LET, as in this program example.

```
10 LET A$="THIS"
```

Notice that you must put quotation marks around the words in a string variable.

String variables can hold any combination of letters, numbers, punctuation marks, or graphics symbols. The next program uses lots of string variables. Before you type it in, see if you can figure out what it will print.

```
10 LET A$="THIS"
20 LET B$=" DOES"
30 LET C$="KE SEN"
40 LET D$=" MA"
50 LET E$="YOU? "
60 LET F$="SE TO "
70 PRINT B$;A$;D$;C$;F$;E$;
```

Now, what does this say?



## Review \*\*\*\*\*

String variables—a variable box that can hold letters or words is called a string variable. Examples are G\$ and Q\$ (G string and Q string).

### INSECTICIDE PATROL #8



This program has *three* bugs. Can you debug it? (Answer on page 128.)

```
10 LET A$=CLARA
20 LET B$="HATES"
30 LET C$="BUGS"
40 PRINT A$
50 PRINT B$
60 PRINT C$
```



## LESSON 4

Many programs, such as games, ask the user to type answers or other information. String variables and the command INPUT are thus very useful. Time now for you to use them in a longer game which will print graphics patterns. We'll start the program in this lesson and finish it in the next lesson.

The first lines set up string variables made of graphics characters. These variables will be used to make patterns later on.

```
10 LET G$="          "
20 LET H$="          "
30 LET I$="          "
```

To type all these strings, you must be in graphics mode (SHIFT and GRAPHICS). Line 10 is just eight spaces. For line 20, type a space and then graphics A (SHIFT and A), four times. Line 30 is graphics T then graphics Y, repeated four times.

Next, we'll have Clara ask the user's name and store it in variable A\$.

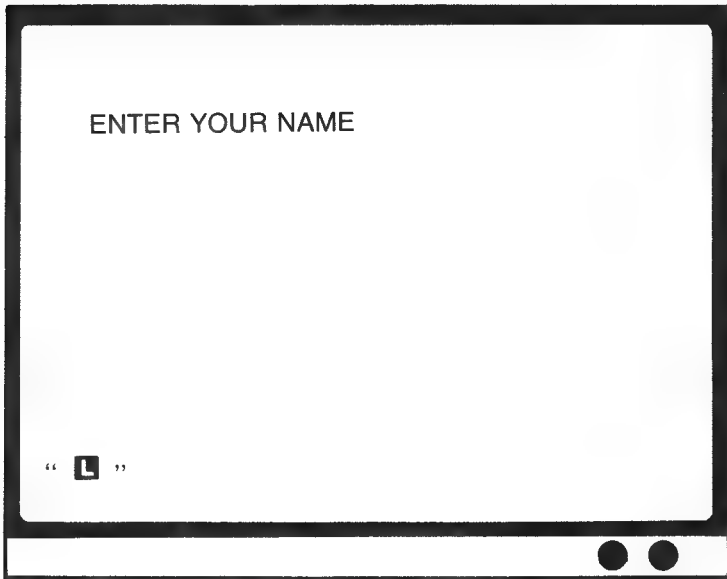
```
40 PRINT "ENTER YOUR NAME"
50 INPUT A$
```

The next part of the program will ask the user to type three favorite kinds of food. All of this data will be used in graphics patterns later on.

```
60 PRINT "ENTER THREE FOODS YOU LIKE."
70 PRINT "FIRST FAVORITE"
```

```
80 INPUT B$
90 PRINT "SECOND FAVORITE"
100 INPUT C$
110 PRINT "THIRD FAVORITE"
120 INPUT D$
130 CLS
```

Before we go on with the program, let's do a little experiment. Run the program. When the computer asks you for your name, the screen will look like this.



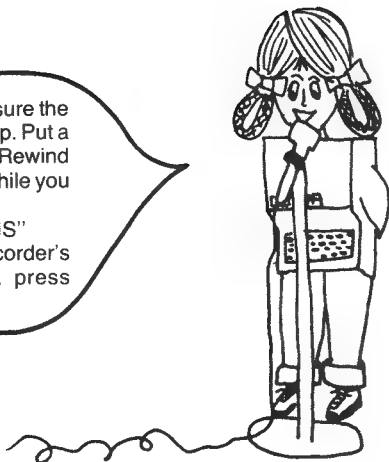
Now, delete the second quotation mark, type your name, and press ENTER. Clara tells you that you made a **S**illy mistake—the missing quotation mark. Sometimes the quotation marks get erased by mistake when entering string input. When this happens, you *must* put the quotation marks back in before Clara will understand your input. Watch those quotation marks!

If you are not going to go directly on to Lesson 5, save this program on tape.

To save this program, make sure the recorder is properly hooked up. Put a usable tape in the recorder. Rewind it, then press Fast Forward while you count 1000, 2000, 3000.

Type SAVE "PATTERNS"

As soon as you press the recorder's Play and Record buttons, press ENTER.



## LESSON 5

On with our long patterns program! If the program isn't still in the computer, load it from the tape recorder.

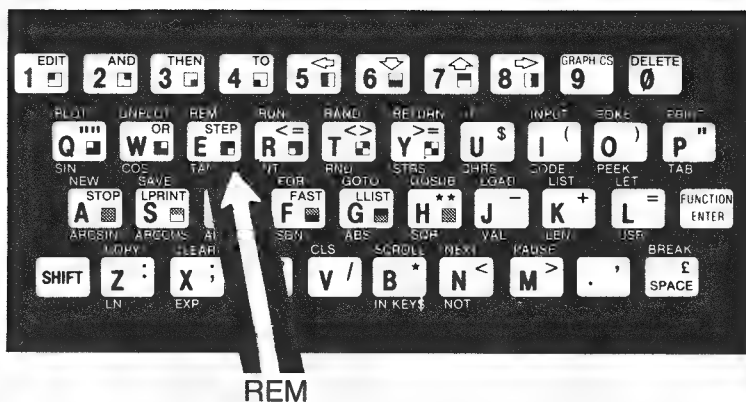
1. After making sure the recorder is connected properly and the volume is set at the right level, type the following:  
**LOAD "PATTERNS".**
2. Hit **ENTER** and the tape recorder's play button at the same time.

When typing a long program like this, use **FAST**. Remember to type **SLOW** before you run the program, though!

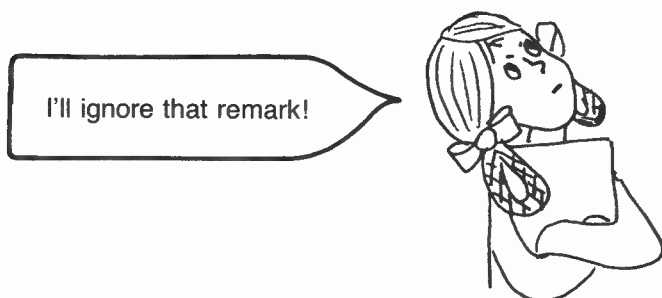


Long programs can be confusing when you list them. To make our program easier to understand, type this line:

```
200 REM FIRST PATTERN
```



REM stands for REMark. Clara will ignore anything you type after the command REM. It is there to help us understand the program better, but it doesn't change how the program runs. REMarks stand out even better if they are inverse video.



```
210  FOR T=1 TO 8
220  PRINT G$;A$;I$;B$;G$;C$;
230  NEXT T
```

This loop will print a pattern made of the string variables. The semicolons in line 220 are very important, so be sure to type them all.

This isn't the only pattern in our program, though. Before we go on to the next one, we should give the user a chance to look at the pattern. Since we'll be pausing and clearing the screen more than once, we'll use a subroutine. Do you remember why we need line 490?

```

240 GOSUB 500
490 STOP
500 REM CLS SUBROUTINE
510 PAUSE 300
520 CLS
530 RETURN

```

Here is the next pattern for you to type. Notice that the order of the string variables is jumbled. As long as we print them in a different order, we'll get a different pattern.

```

300 REM SECOND PATTERN
310 FOR T=1 TO 8
320 PRINT I$;D$;A$;H$;C$;I$;
330 NEXT T
340 GOSUB 500

```

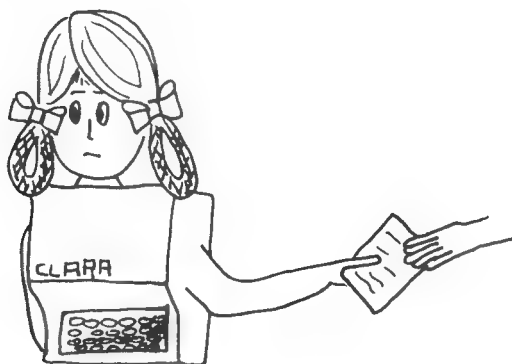
Can you add a pattern or two of your own? Try it!

## Review \*\*\*\*\*

REM—tells the computer to ignore whatever is typed in the rest of the line. REMarks can help you to make a program listing easier to understand.

## LESSON 6

Have you ever passed a note to a friend—the kind of note that says “Elizabeth loves Brett?” Then Game 5 is for you. Clara will be passing love notes!



We will also be studying the Game 5 program as an example of how a longer program is put together.

1. Set up the tape recorder with the correct volume level.
2. Put in and rewind the Chapter 5 tape. (See page 143.)
3. Type LOAD "GAME 5"
4. Hit ENTER and play at the same time.
5. Play the game. Then come back to this page. You can learn a lot from studying this program.

\* \* \*

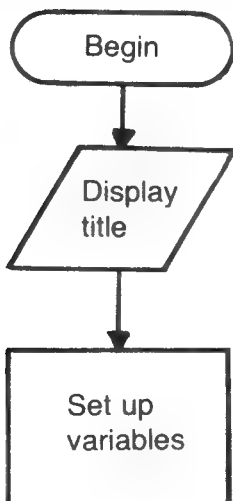
As we study the program, I'll show you flowcharts of the part of the program you're learning about. See if you can find the program lines that go with the flowchart.

Type LIST, and look at line 60. Here we see a new kind of string variable, called an *array*. You can store more than one name in an array by using subscripts. For example, in lines 60 to 90, Game 5 uses A\$(1), A\$(2) and A\$(3). The numbers 1, 2 and 3 are the subscripts.

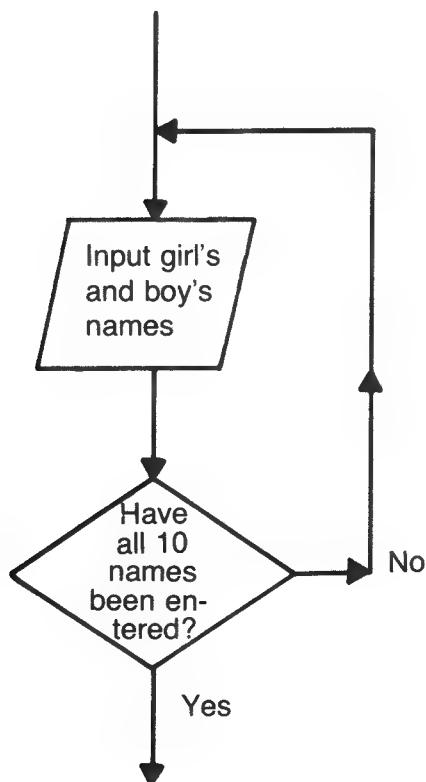
Notice also line 50.

**DIM A\$ (3,5)**

The 3 tells Clara how many A strings there will be. The 5 tells Clara how many letters long the A strings can be. Line 50 makes the computer set aside enough space in memory for the array.

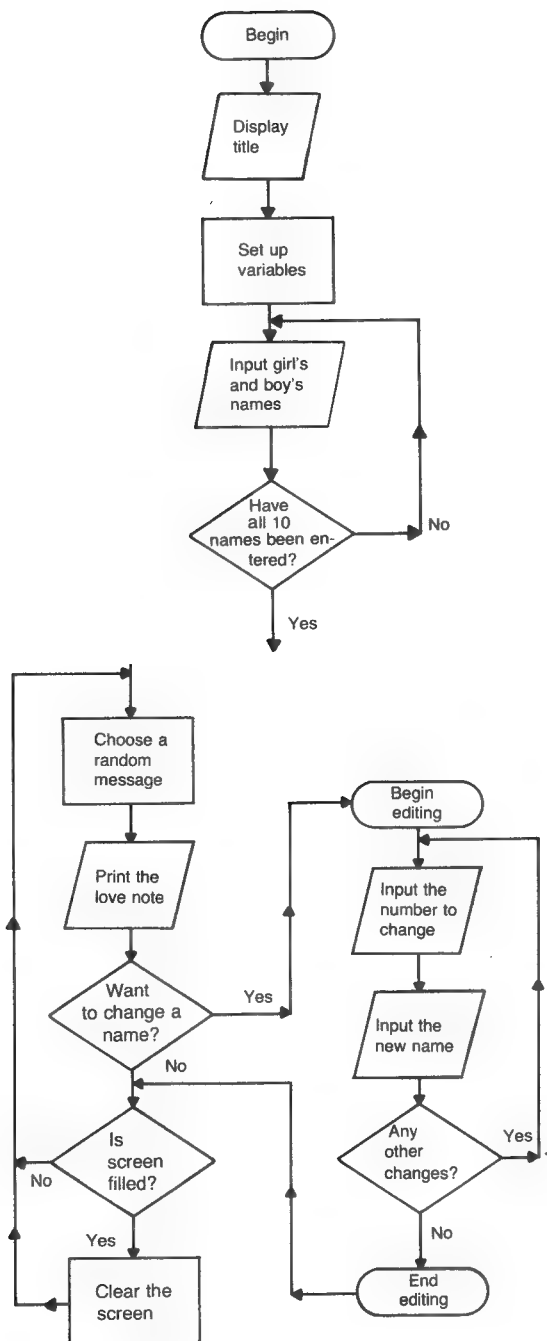


The names are input during a FOR . . . NEXT loop that is repeated ten times, as shown in the flowchart. The names are stored in another array N\$. Can you find the line number that DIMensions, or sets aside memory space, for N\$?



Look at the next section of the program. Notice that in lines 170 and 180 we use the regular variable N as a subscript—N\$(N). The first time through the loop, N equals 1 and N\$(N) is N\$(1). The second time through the loop, N equals 2 and N\$(N) is N\$(2), and so forth.

Below is the complete flowchart for Game 5. It looks complicated at first, but follow the arrows to see how the program works. Then see if you can figure out the line numbers that go with parts of the flowchart. For example, at what line number does the Editing Subroutine begin? How are subscripts used to let Clara know which name to edit? Which line numbers choose the random message?





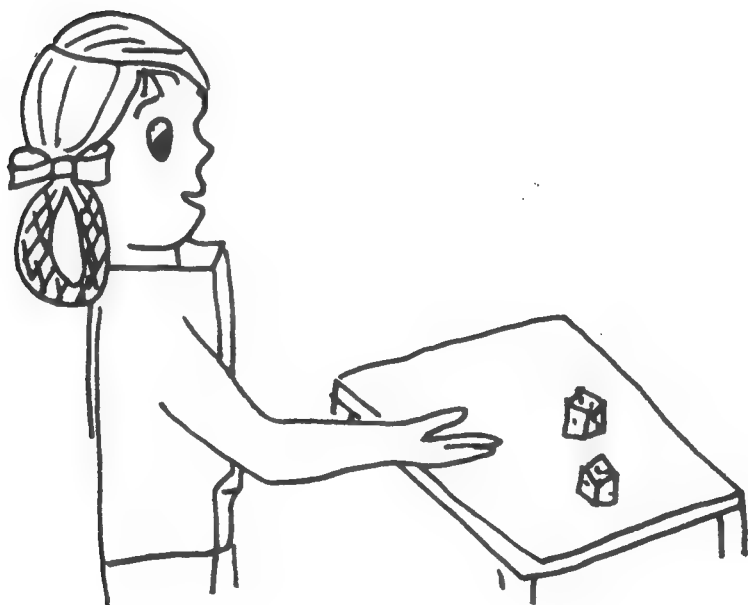
## Review \*\*\*\*\*

DIM—tells the computer to DIMension, or set aside memory space, for an array. For example, DIM F\$(8,10) sets aside space for eight F strings that can be up to ten letters long.

## LESSON 7

Do you like games? A good game needs two things—skill and luck. Take Monopoly, for example. You need skill to decide what properties to buy and what not to buy, when and where to put houses and hotels, etc. You also need luck so that the numbers on the dice don't land you on Boardwalk with your friend's four houses!

Board games like Monopoly use dice, or spinners, or cards to put luck into a game. The dice produce a bunch of *random* numbers—you never know what number will come up next. Clara can also produce random numbers, to put luck into a game. Try this program, which makes Clara act like dice—almost.

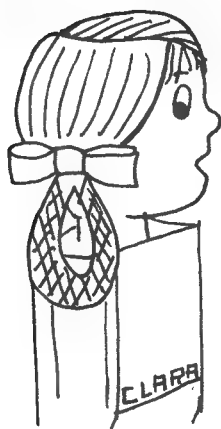
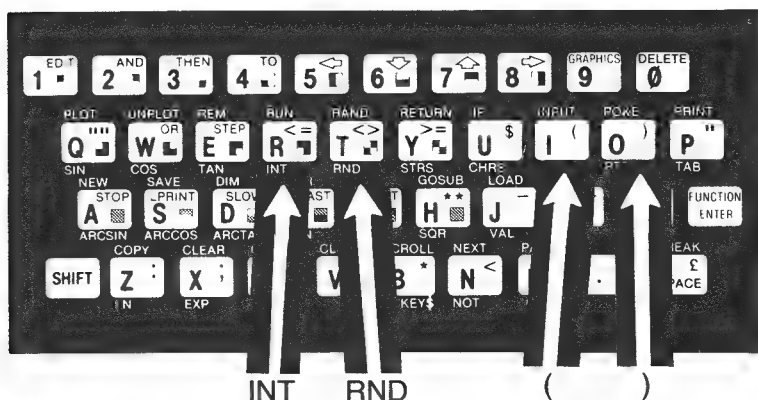


```

10  SCROLL
20  LET A=INT (RND *6)
30  PRINT A
40  GOTO 10

```

When you run this the screen will fill with numbers between zero and five - but you won't know what number comes next, because they are random. To stop the program, press BREAK.



Since INT and RND are functions, written below the keys, remember to hold SHIFT while you press ENTER to make the screen show **F**.

Did you see one of the reasons I said this is *almost* like dice? Dice are numbered from one to six, not zero to five. The line INT (RND \*6) picks one of six random numbers—0, 1, 2, 3, 4 or 5. To make Clara pick a number between one and six, change line 20 to this, then run the program.

```
20 LET A=INT (RND*6)+1
```

RND stands for RaNDom, of course. INT stands for INTEger. To discover what INT does, retype line 20 again, but leave out INT. Run the program. Weird, isn't it? You can see why we usually use INT with RND, to get rid of all those decimal places!

**Review** \*\*\*\*\*

INT (RND \* number)—tells the computer to pick a number at random, like dice or a spinner. INT (RND \*10) will choose a number between 0 and 9. INT (RND \*10)+1 will choose a number between 1 and 10.

### INSECTICIDE PATROL#9

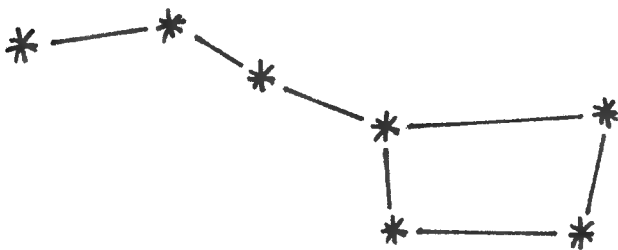


This program has two bugs. Can you debug it? (Answer on page 128.)

```
10 REM "CLARA ";  
20 PRINT "HATES WHAT?"  
30 INPUT BUGS$  
40 FOR T=1 TO 30  
50 PRINT BUGS$,  
60 NEXT T
```

## LESSON 8

Can you find the Big Dipper in the night sky?



Planets of other stars would have different constellations, or patterns of stars. We can use RND to make Clara show us the stars as they might appear on another planet. Here is the program.

```
10 REM STARRY NIGHT
20 FOR T=1 TO 50
30 LET A= INT (RND *22)
40 LET B= INT (RND *32)
```

This sets up a loop that will print 50 stars. Line 30 picks a number from 0 to 21—the number of rows on the screen. Line 40 picks a number from 0 to 31—the numbers of the up and down columns on the screen.

```
50 PRINT AT A,B;"*"
60 NEXT T
```

Line 50 actually prints a star at a random place on the screen. Line 60 finishes the loop, then goes back to line 20 to print another star in another random place.

When you run the program, see if you can find any star patterns, or constellations, in the sky of our imaginary far distant planet.



## LESSON 9

Now that we have *seen* the stars of outer space, let's blast off to *visit* the stars!

We will write a program to draw a spaceship and then launch it upwards and outwards. Begin with a title for the program in REM statement—a good programming habit to develop.

### 10 REM SPACESHIP LAUNCH

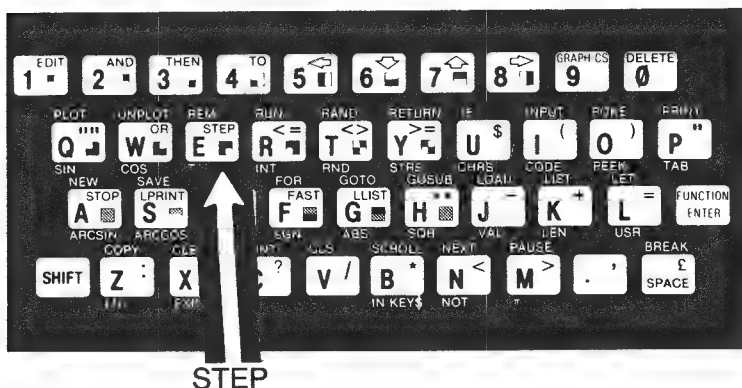
Would you like some stars in the sky behind the spaceship? If so, use INT (RND \* number) to scatter 10 or 15 stars.

The next part of the program is the spaceship itself, which I am going to leave up to you. Design and program your ship so that it sits on the bottom line of the screen, row 21. You'll need to use PRINT AT.

It makes sense that we should have a button to push to launch the spaceship. INPUT is a handy command for this, as Clara waits till the user presses ENTER before continuing the program. We won't use the data from the input; it is only a means of beginning the launch. Add these lines to your program.

```
200 PRINT AT 0,0;"PRESS ENTER TO LAUNCH"  
210 INPUT A$
```

What is a launch without a countdown? The next few lines make Clara count backwards like a real countdown. Notice how lines 240 and 250 set up an empty loop that just kills time so the countdown doesn't go too quickly.



```

220  FOR T=10 TO 0 STEP -1
230  PRINT " ";T;
240  FOR P=1 TO 8
250  NEXT P
260  NEXT T

```

The launch itself will use `SCROLL` and another `FOR . . . NEXT` loop.

```

270  FOR T=1 TO 22
280  SCROLL
290  NEXT T

```

There is the program. Many happy landings!

### Review \*\*\*\*\*

`STEP`—this command tells the computer to count differently than usual in a `FOR . . . NEXT` loop. Examples:

```

FOR Q=100 TO 200 STEP 5
FOR T=100 TO 0 STEP -10

```

### INSECTICIDE PATROL #10



This program has a bug. Can you find it? (Answer on page 129.)

```
10 LET A=INT (RND *100)+10
20 LET A$="CLARA HATES BUGS"
30 SCROLL
40 FOR T=1 TO A STEP 2
50 PRINT A$
60 NEXT T
```

## LESSON 10

Quiz time once again! Do you need to review anything before you take the quiz?

Set up and load the program as you have done before. The name is "QUIZ 5" and it is on the Chapter 5 tape. (See page 144.)

Quiz 5 works the same as the other quizzes. Good luck!



## QUIZ 5

1. Which of these BASIC commands tells Clara to let someone put in data while a program is running?
2. One of these commands tells Clara to ignore the rest of the program line. Which one?
3. A program is running. You decide you want to do something else, so you stop the program before it is finished. What key should you press?
4. Which of these commands *cannot* be used with a string variable?
5. Which of these lines will choose a random number between 1 and 9?
6. Which of these lines *might* choose the number 0?

7. Which of these lines *might* choose the number 4.73?
8. You want to use an array of string variables of the kind B\$(1), B\$(2), B\$(3). Which of these lines correctly sets aside the memory needed?

Study this program, then answer the questions about it.

```
10 PRINT "WHAT IS YOUR NAME?"
20 _____
30 PRINT "HOW OLD ARE YOU?"
40 _____
50 PRINT "HELLO, ";A$;" , HOW ARE YOU?"
60 LET B= INT (RND *5)+5
70 PRINT "I AM";B;" YEARS OLD."
```

9. Which of these BASIC statements belongs in line 20 of the program?
10. Which of these BASIC statements belongs in line 40 of the program?
11. How many string variables does this program use?
12. According to this program, which of these numbers *might* be the age of the computer?



## Chapter 6

# Decisions, Decisions



### LESSON 1

Here is a 'program' that your mother might give to you one day.

```
10 GET READY FOR SCHOOL
20 GOTO SCHOOL
30 IF TIME = 3:00 THEN GOTO HOME
40 IF TIME > 10:00 THEN GOTO BED
50 GOTO 10
```



A lot of GOTO's! Of course, since your mother speaks English rather than BASIC, she would use go to. .

Notice also the way time is used to decide whether you do certain things, as in line 30 and line 40.

Telling the computer to GOTO bed or GOTO school is rather a waste of time, since it can't walk. Using GOTO in programs, though, can be very useful. Just to see how well you understand GOTO, try to figure out what this confusing program will print.

```
10 GOTO 60
20 PRINT "IS"
30 GOTO 80
40 PRINT "NICE"
50 GOTO 100
60 PRINT "CLARA"
70 GOTO 20
80 PRINT "A"
90 GOTO 40
100 PRINT "COMPUTER"
```

It's your turn now. Can you use a lot of GOTO's in a program that will print this?

THE PERSON  
I WOULD MOST  
LIKE TO KISS  
IS (anyone you choose).

#### INSECTICIDE PATROL #11



This program has a bug in it. Can you debug it? (Answer on page 129.)

```
10 GOTO 60
20 PRINT "BUGS "
30 GOTO 80
40 PRINT "HATES "
50 GOTO 100
60 PRINT "CLARA "
70 GOTO 40
80 STOP
```

## LESSON 2

Off to the candy shop, coins jingling in your pocket—what to buy?



Pecan log roll is your all-around favorite, but it costs \$3. If you don't have enough money for the log roll, you decide to get an all-day sucker instead. In other words, IF your money is greater than \$3, THEN you will buy a log roll. Your decision depends on how much money you have.

After the candy shop, you have homework to do. How will you decide when to stop your homework and go outside to play? When the last question is done, of course! In other words, IF your homework is done, THEN you can go play.

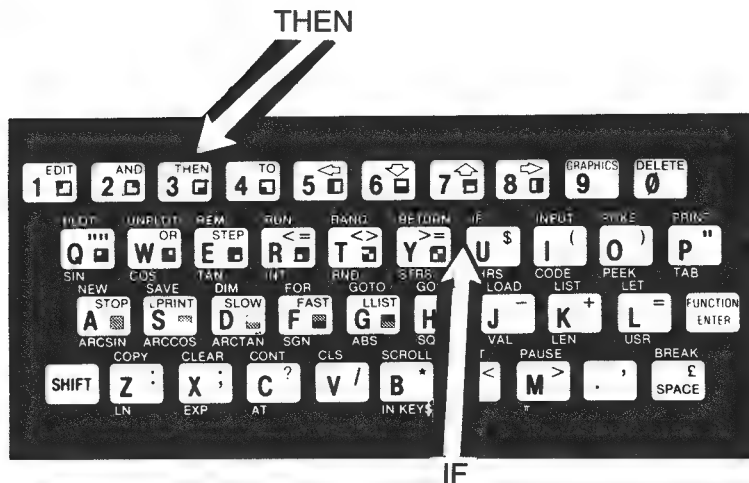


Clara can make decisions in much the same way, using the keywords IF and THEN. For example, type the following program. The computer will 'decide' to keep printing random numbers if it hasn't yet printed 15 of them.

```

10 LET COUNT=0
20 LET R=INT (RND * 1000)
30 PRINT R
40 LET COUNT=COUNT+1
50 IF COUNT<15 THEN GOTO 20

```



Write an IF...THEN program of your own that will count from 1 to 44. Use a comma after your print line so the screen will look like this:

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44

## Review \*\*\*\*\*

IF...THEN—the computer uses these commands to ‘decide’ what to do.

## LESSON 3

You have to make a lot of decisions every day—what to wear, what to do, and so forth. Computers make lots of decisions, too, so that IF...THEN is very important. Let’s see how we can make Clara keep score in a math game.



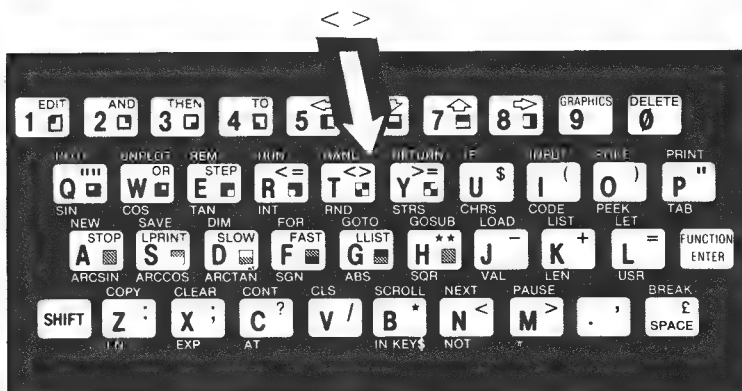
To do this, Clara will have to decide when to add points. Type this program.

```
10 REM MATH GAME
20 LET SCORE=0
30 FOR T=1 TO 10
40 LET A=INT (RND * 10)
50 LET B=INT (RND * 10)
60 PRINT A; " X "; B
70 INPUT I
```

Line 20 makes the score zero at the beginning of the game. Line 30 determines that there will be ten problems in the game. Lines 40 to 70 come up with a random times table problem, print the program, and then wait for the user's answer.

```
80 IF I=A*B THEN GOSUB 200
200 REM SCOREKEEPING SUBROUTINE
210 LET SCORE=SCORE+1
220 PRINT "CORRECT"
230 RETURN
```

Line 80 decides that if the input is correct the computer will go to a subroutine at line 200. The subroutine tells Clara to increase the score and print that the input was right.



```
90 IF I<>A*B THEN PRINT "WRONG"
100 NEXT T
```

Do you remember what the sign  $< >$  means? That's right—not equal to. If the input is not equal to the correct answer, the computer will tell you so. Since the program never got to subroutine 200, the score is not increased. Clara then goes on to the next problem.

```
110 CLS
120 PRINT "YOU GOT ";SCORE;" OUT OF 10."
130 STOP
```

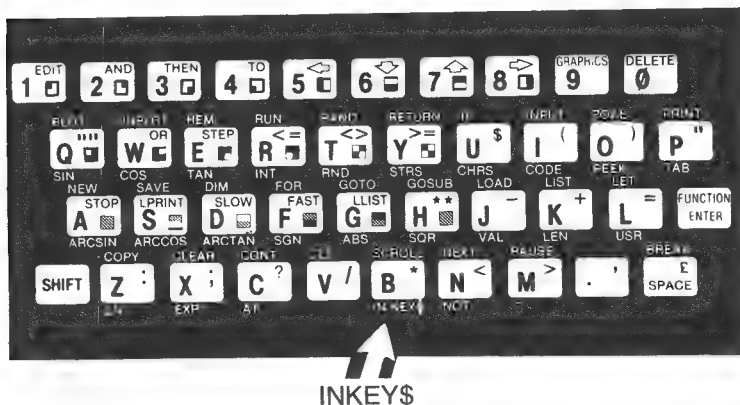
Lines 110 to 120 make the computer show the score. Line 130 keeps Clara from crashing into the subroutine at line 200.

How many problems can you get right?

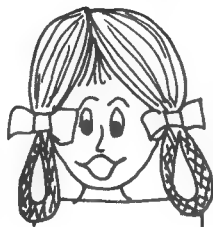
## LESSON 4

Some computer programs let you enter input without pressing ENTER!

Games 1 and 4, where you pressed the key that was flashed on the screen, are examples. The secret is the BASIC word INKEY\$, pronounced in-key-string. To see how INKEY\$ works and how it differs from INPUT, type the following program. After the equals sign in line 10 are two quotation marks *without* a space in between. Hold SHIFT while you press P twice.




INKEY\$ is a function. Do you remember how to make the screen show **F**?



```
10 IF INKEY$ = "" THEN GOTO 10
20 PRINT INKEY$;
```

Unlike INPUT, which makes the computer wait for you to type your answer, INKEY\$ *doesn't* wait. If no key is pressed the computer just charges right on to the next program line. Thus, line 10 says to Clara if no key is pressed (nothing between the quotation marks, remember) then keep going back to line 10 until a key *is* pressed.



With INKEY\$, I charge right on to the next program line!

Now run the program. Line 10 makes Clara wait until a key is pressed and line 20 prints the key you first touch. Here is another important difference: INKEY\$ makes the computer pay attention to only one letter, and not a whole word like INPUT does. To make our program print more than just one letter, add the next line.

```
30 GOTO 10
```

Wow! When you type something now you sure can fill the screen quickly! If the screen fills, press CONT to continue.



We can also use INKEY\$ to stop the program whenever we wish. Add this line, being sure to put quotation marks around the A.

```
25 IF INKEY$="A" THEN STOP
```

Try some more typing. What happens when you press the A key? Add a line 26 that will use INKEY\$ to make Clara clear the screen when you press the V key.

**Review** \*\*\*\*\*

INKEY\$—tells the computer to pay attention to the key that is being pressed.

## LESSON 5

The last program was fun the way it acted like an electric typewriter. If the program isn't still in the computer's memory, re-enter it, because we're going to work on it some more.

```
10 IF INKEY$="" THEN GOTO 10
20 PRINT INKEY$
25 IF INKEY$="A" THEN STOP
26 IF INKEY$="V" THEN CLS
30 GOTO 10
```

First, let's add a title line to our program.

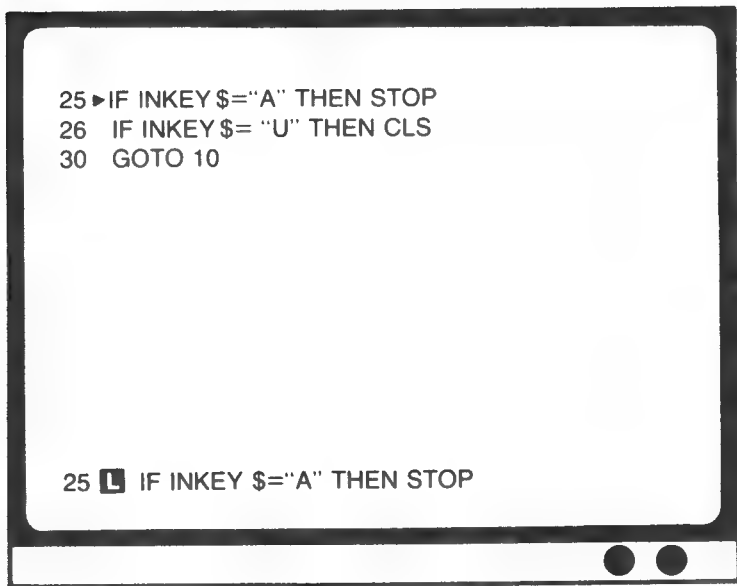
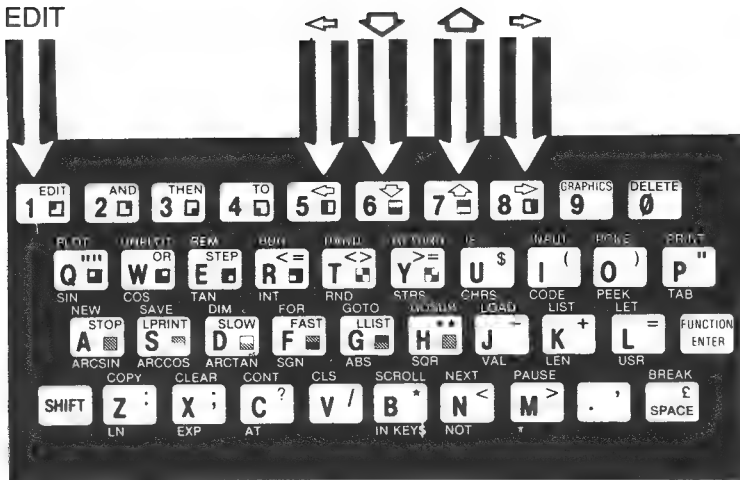
```
1 REM TYPEWRITER
```

Now, let's suppose that you want to be able to type the letter A without stopping the program. We'll use the period key to stop instead.

One way to make this change would, of course, be to type a whole new line 25. There is an easier way, though.

1. Type LIST 25. Notice that the black arrow points to line 25.
2. Hold SHIFT while you press EDIT.
3. Line 25 now appears at the bottom of the screen as well as the top. We are now ready to begin editing, or changing, this line.
4. Hold SHIFT while you press the arrow that points to the right.
5. Keep pressing the 'right' arrow key until the **L** is to the right of the letter A.

EDIT



25 IF INKEY\$="AL" THEN STOP

6. Delete the letter A as usual. Retype a period instead, and press ENTER. The edited line 25 replaces the old line.



EDIT can save you a vast amount of time!

Now press the 'up' arrow key. (Did you remember to hold SHIFT? Notice that (■), called a cursor, has now moved up to point at line 20. Now you would be ready to begin editing line 20. Press the 'down' arrow twice and the cursor point to line 26. The 'up' and 'down' arrows are used to move the cursor to the line you want to edit.

The 'left' and 'right' arrows are used after you have already pressed EDIT to move back and forth in a line without having to erase everything. You can use them when you are typing a program line also, not just when editing.

#### **Review** \*\*\*\*\*

EDIT—tells the computer to change a program line. The line the cursor points to moves to the bottom of the screen for editing.

Arrow keys—these keys move the cursor through your program to the exact part you want to change.

#### **LESSON 6**

Game time again! This time our game will scramble a word and ask you to figure out what the word is. The words to be scrambled are mostly computer words. After you have tried the program, come back to this page to learn how to edit it to put in words of your own.

The program name is "GAME 6", on the Chapter 6 tape. (See page 146.)

\* \* \*

Were you able to unscramble many of the words? You might want to use this program with words of your own. Even if you don't, you should read the rest of this lesson to learn more about editing.

Type FAST before you begin to edit.  
This speeds things up even more!



1. Type LIST. Notice that the words are stored in an array, W\$( ). Later in the program, line 220 chooses a random number which is put into the subscript. This determines the word to be scrambled.
2. Type LIST 80. This is an easy way to move the cursor to line 80. Hold SHIFT while you press EDIT.
3. Use the 'right' arrow to move the **L** past the word programmer.

```
80 LET W$(1)="PROGRAMMERL"
```

4. Delete the word programmer. Type your new word, no more than ten letters long. Then ENTER.

Are the quotation marks still in place?



5. Use the 'down' arrow to move the cursor to line 90. Edit line 90 as above and put in your own word.
  6. You can put more words of your own into the other parts of the array in lines 100 to 170.
  7. Press SLOW before you run the game.
- Happy scrambled words!

## INSECTICIDE PATROL #12



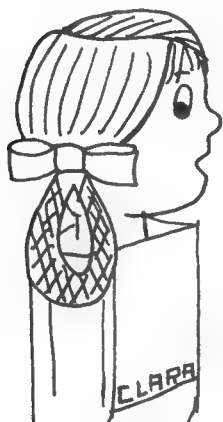
This program has a bug in it. Can you debug it? (Answer on page 129.)

```
10 PRINT "WHAT DOES CLARA HATE?"
20 INPUT B$
30 IF B$ <> "BUGS" THEN GOTO 60
40 PRINT "NOPE — TRY AGAIN."
50 GOTO 20
60 PRINT "THAT IS RIGHT, "; B$
```

## LESSON 7

Home runs, stolen bases, World Series...ah, baseball! Did you know that Clara can be useful for a fan or player? Type the following

program, which calculates either batting averages or a team's won-lost percentage. Notice how IF...THEN is used to decide which part of the program to run (lines 60 to 80).



Since this is a longer program, you might find it quicker to use FAST while programming.

```
1  REM BASEBALL CALCULATIONS
10 PRINT "WHAT DO YOU WANT TO DO?"
20 PRINT "TYPE THE NUMBER."
30 PRINT "1. BATTING AVERAGES"
40 PRINT "2. WON-LOST PERCENTAGES"
50 PRINT "3. STOP"
60 IF INKEY$="1" THEN GOTO 100
70 IF INKEY$="2" THEN GOTO 190
80 IF INKEY$="3" THEN STOP
90 GOTO 60
```

Lines 10 to 50 print the menu for this program. Lines 60 to 80 let the user decide what part of the menu he wants to use. Line 90 is necessary because INKEY\$ doesn't wait for input. If the user hasn't typed a number yet, line 90 keeps going back, making Clara wait. Lines 60 to 90 are also neat because they won't let the user type a 'wrong' answer, like L or 6. Anything other than 1, 2 or 3 is ignored!

```
100 REM BATTING AVERAGES
110 CLS
120 PRINT "NUMBER OF AT BATS?"
130 INPUT A
140 PRINT "NUMBER OF HITS?"
150 INPUT H
```

```

160 PRINT
170 PRINT "BATTING AVERAGE IS .";INT(H/A)*1000
180 GOTO 10

```



This part of the program lets the user put in data about the number of hits and at-bats. Line 170 actually figures out the batting average.

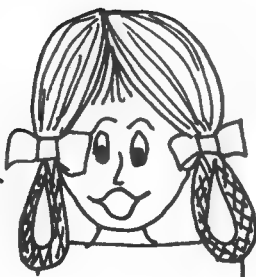
Now finish the program by typing in the last routine.

```

190 REM W-L PERCENTAGE
200 CLS
210 PRINT "NUMBER OF WINS?"
220 INPUT W
230 PRINT "NUMBER OF LOSSES?"
240 INPUT L
250 PRINT
260 PRINT "W-L PERCENTAGE IS .";INT (W/
    (W+L))*1000
270 GOTO 10

```

Run the program and input the following numbers, or use numbers of your own.



### *Batting Averages*

George Brett, 1980	449 at bats, 175 hits
Hank Aaron, 1957	615 at bats, 198 hits
Reggie Jackson, 1973	539 at bats, 158 hits

### *Won-Lost Percentages*

1927 Yankees	110 wins, 44 losses
1959 White Sox	94 wins, 60 losses
1974 Dodgers	102 wins, 60 losses

## **LESSON 8**

Are computers magic? By now, of course, you should know they aren't. Computers just do what they are programmed to do. You can, however, use what you have learned to teach Clara a 'magic' trick. Type the following program.

```
10 REM MAGIC TRICK
20 PRINT "I CAN DO A MAGIC TRICK."
30 PRINT
40 PRINT "THINK OF A NUMBER, BUT DONT"
50 PRINT "TELL ME WHAT IT IS."
```

At this point and at other points in the program we will give the user time to think of a number before we go on. Since we will do it several times, let's put it in a subroutine.

```
60 GOSUB 300
300 REM READY? SUBROUTINE
```



```

310 PRINT " ARE YOU READY? (Y OR N)"
320 IF INKEY$ < > "Y" THEN GOTO 320
330 PRINT
340 RETURN

```

Next the computer asks the user to do some arithmetic with his number. After each time, GOSUB 300 will give time to do the math.

```

70 PRINT "MULTIPLY YOUR NUMBER BY 8."
80 GOSUB 300
90 PRINT "ADD 6 TO THE ANSWER."
100 GOSUB 300
110 PRINT "NOW DIVIDE THE ANSWER BY 2."
120 GOSUB 300
130 PRINT "SUBTRACT 3."
140 GOSUB 300

```

Now Clara is ready to do her trick.



```

150 PRINT "THEN INPUT YOUR FINAL ANSWER."
160 INPUT A
170 LET B=((A+3)*2-6)/8
180 CLS
190 PRINT "YOUR NUMBER IS",B
200 STOP

```

Line 170 is the key to the trick. Can you see how it undoes all the user's math—adds 3 instead of subtracts 3, and so on?

## INSECTICIDE PATROL #13



This program has a bug. Can you debug it? (Answer on page 129.) This bug is tricky, so you might need to type it in to find the problem.

```
10 LET P=1
20 DIM A$(3,5)
30 LET A$(1)"CLARA"
40 LET A$(2)="HATES"
50 LET A$(3)="BUGS"
60 LET P=P+1
70 IF INKEY$="" THEN GOTO 70
80 PRINT A$(P)
90 GOTO 60
100 IF P=3 THEN STOP
```

## LESSON 9

You've almost finished the book!

You've learned a lot, I hope. You now know most, though not all, of the BASIC words that Clara uses. To see just how many things you have learned, let's list them.

PRINT	NEW
ENTER	RUN
LOAD	CLS
SAVE	PRINT AT
GRAPHICS	DELETE
PAUSE	GOTO
LET	CONT
FAST	SLOW
FOR...TO	NEXT
STOP	LIST
GOSUB	RETURN
SCROLL	INPUT
INT	RND
EDIT	IF...THEN
REM	DIM

This sure is an impressive list of commands!



How many of these BASIC words can you use in one program? Write a game program of your own, using as many of them as you can. Since games need luck as well as skill, use `INT(RND*number)`. Use `INPUT` or `INKEY$` to let the player 'talk' to the computer.

Good luck!

## LESSON 10

Your last computer quiz!

Do you understand all the BASIC words in the last lesson? If not, perhaps you should review those commands you are unsure of before you take the quiz.

The program name is "QUIZ 6" on the Chapter 6 tape. (See page 147.)



### QUIZ 6

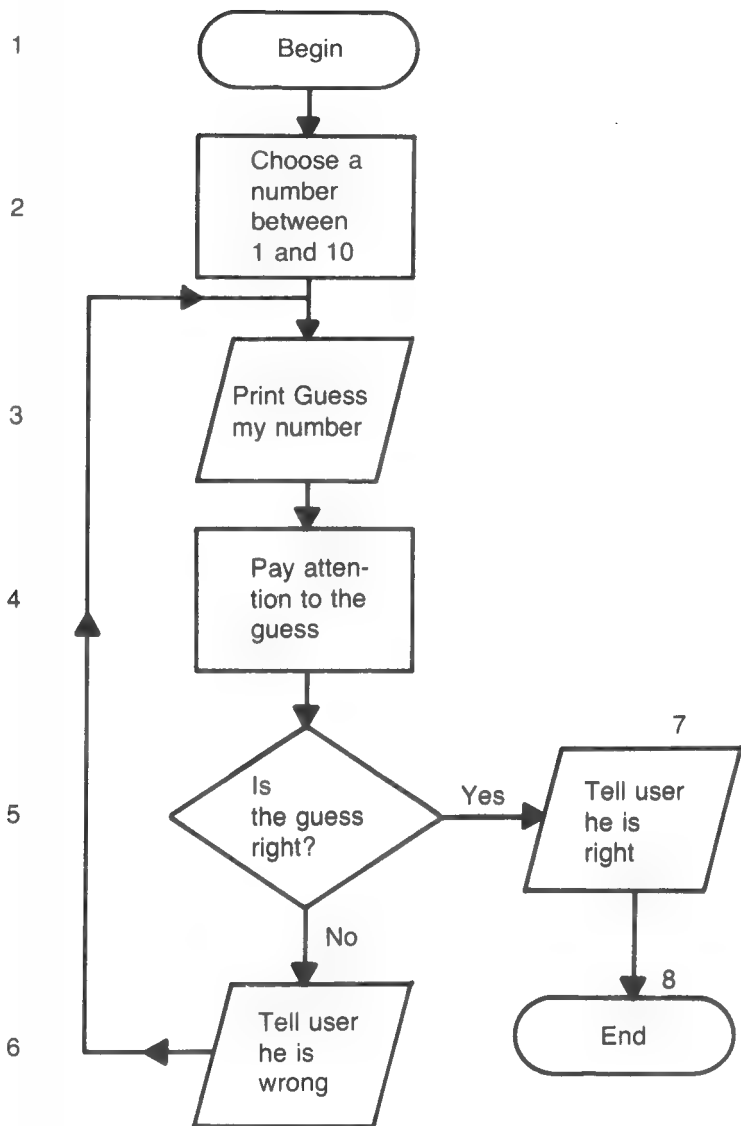
1. You want to change a program line. The cursor already points to this line. What must you press before you can begin making changes?
2. Which of these shapes is used as a cursor during editing?
3. Which arrow can be used to move the cursor to a program line you want to edit?
4. Which arrow key can be used to move backwards within a program line without having to delete anything?

Study the flowchart on the next page, and then answer the questions about it.

5. Which program line goes with step 6 of the flowchart?
6. Which line of BASIC goes with step 2 of the flowchart?
7. Which line of BASIC goes with step 4 of the flowchart?
8. Which line of BASIC goes with step 5 of the flowchart?

Study the program on page 126. Then answer the questions about it.

9. Which program line belongs in line 10?
10. Which program line belongs in line 50?
11. Which program line belongs in line 70?
12. Which program line has a bug in it?



```
10  REM LOVERS
20  PRINT "DO YOU LOVE ME?"
30  IF INKEY$="Y" THEN GOTO 90
40  IF INKEY$="N" THEN GOTO 60
50  -----
60  PRINT AT 5,0; "SAME TO YOU, BU
D."
70  -----
80  PRINT AT 5,0; "I LOVE YOU, TOO
"
```



## Insecticide Patrol Answers

---

#1. *Line 20:* The second quotation mark was left off.

```
20 PRINT "HATES"
```

#2. *Line 10:* The second quotation mark should not be inverse video.

```
10 PRINT "CLARA"
```

#3. *Line 10:* A semicolon is needed after the row and column numbers.

```
10 PRINT AT 18,5;"CLARA"
```

*Line 30:* The column number, 32, is off the screen. Also, there must be at least four columns to print the word BUGS without going off the screen.

```
30 PRINT AT 20,28;"BUGS"
```

#4. *Line 40:* PAUSE must have a number after it to tell Clara how long to wait.

```
40 PAUSE 100
```

**#5. Line 90:** There must be a RETURN at the end of every subroutine.

90 RETURN

**#6. Line 10:** The loop goes past 21, which is the last row on the screen that line 20 can print at.

10 FOR T=1 TO 21

*Line 40:* There must be a NEXT . . . line at the end of every loop.

40 NEXT T

**#7. Line 30:** The computer will never get past this line because of the GOTO. Delete the whole line by typing 30 and then ENTER.

*Line 50:* We cannot use variable B before we have told Clara to make a box named B. Commands that set up variable boxes are LET, INPUT and FOR. In this case, we need LET. The line can go anywhere in the program before line 50.

5 LET B=100

**#8. Line 10:** String variables must have quotation marks around the word.

10 LET A\$="CLARA"

*Line 20:* The string variable is mistyped.

20 LET B\$="HATES"

*Line 30:* C is a regular variable and it should be a string variable.

30 LET C\$="BUGS"

**#9. Line 10:** Clara ignores REM statements. Rewrite it as a PRINT statement.

10 PRINT "CLARA ";



*Line 30* and *Line 50*: String variable names can only be one letter long.

```
30 INPUT B$  
50 PRINT B$
```

**#10.** *Line 30* and *Line 40*: We must SCROLL at the beginning of each loop. Thus, line 30 needs to be moved into the loop. One way to do this is to type the following.

```
30  
45 SCROLL
```

**#11.** *Line 50*: This line makes Clara go to line 100, and the program only goes up to line 80. Where we need to go is to the line that will print the word BUGS.

```
50 GOTO 20
```

**#12.** *Line 30*: We should go to line 60 if the input *is* BUGS, not if it *isn't*.

```
30 IF B$="BUGS" THEN GOTO 60
```

**#13.** *Line 60* or *Line 10*: Line 60 makes P equal to 2 the first time through the program. Thus line 80 will never get to A\$(1). Either of these changes will get rid of the bug.

```
10 LET P=0
```

or

```
60  
85 LET P=P+1 (ENTER)
```

# Tape Recorder Use

---

## Connecting a Tape Recorder to the Computer

1. Find the two-wired cord that came with the computer. Notice that the cables are colored differently.
2. On the recorder, put one jack into the 'microphone' socket and the other jack into the 'headphone' socket.
3. On the computer, attach the 'microphone' wire into the socket marked 'MIC' on the side of the computer. Push the other jack into the socket marked 'EAR'.
4. Make sure that you have the wires connected correctly and pushed all the way into the sockets.
5. If the recorder has a tone control, turn it to maximum treble. This will make the tape sound even more 'screechy.'
6. Turn the tape recorder volume control about three quarters of the way up.

## If the Tape Doesn't Load:

1. Did you type the program name correctly?
2. Did you rewind the tape to the correct place?
3. Examine the colors of the cables between the computer and recorder. Is the recorder correctly attached?
4. Is the volume setting correct? Adjust the volume level up or down a little bit (usually up) and try again. When you find a level that works, remember it!

5. Remove all but one of the cables between computer and recorder. In other words, leave the MIC cable in if you are going to SAVE, or leave the EAR cable in if you are going to LOAD.
6. If you still have problems after trying all these ideas, the tape itself might be at fault. Do other tapes load correctly? If not, the tape is probably not at fault. Check the manual that came with the computer for further suggestions.

Usually, once you find the correct setup for your recorder you will have few problems. So keep trying!

# Program Listings

---

All programs run on 2K Timex Sinclair 1000.

## CHAPTER 1

The programs mentioned in the text provide additional practice on computing skills. The six games and six quizzes all will run on a 2K Timex Sinclair 1000. They are available on preprogrammed cassettes. See page 159 for details.

The program listings follow. Take care, for they must be entered exactly as printed. Lines with inverse video or graphics characters are often difficult to figure out. If you have difficulty with any of these lines refer to the program notes, beginning on page 149, for help.

### Game 1

```
10 REM GAME 1
20 REM THE HIDDEN BALL
30 LET T=9
40 LET U=T/T
50 LET H1=NOT U
60 LET H2=H1
70 PRINT AT T,T; " "; "A";
T T+U,T; " " GAME 1 " "; AT 11,T; " "
80 PAUSE 100
90 CLS
100 PRINT TAB T; "DIRECTIONS"
110 PRINT "A LETTER WILL FLIP
SH ON THE", "SCREEN. AS FAST AS Y
```

```

OU CAN,","PRESS THE CORRECT KEY.
120 PRINT " THE FASTER YOU ARE,
THE MORE","POINTS YOU GET."
130 PRINT ","HIT ENTER WHEN ""R
EADY""
140 INPUT R$
150 CLS
160 LET S=NOT U
170 LET B=15
180 PRINT AT U,15;"SCORE"
190 PRINT AT 4,15;"HIGH SCORES"
;AT 5,15;"1";H1;AT 6,15;"2. ";
H2
200 LET B=B-U
210 PRINT AT U,U;"READY?"
220 INPUT R$
230 REM SCORE
240 PRINT AT U,U;" "
250 LET R=INT (RND*36)+28
260 FOR T=70 TO U STEP -U
270 PRINT AT U,U;CHR$ R
280 IF INKEY$="" THEN GOTO 0320
290 IF INKEY$=CHR$ R THEN GOTO
0360
300 PRINT AT U,U;"SCORE"
310 LET T=INT (T/2)
320 PRINT AT U,U;" "
330 NEXT T
340 PRINT AT U,U;"TIME"
350 PAUSE 75
360 LET S=S+T
370 PRINT AT U,21;S
380 IF NOT B THEN GOTO 0400
390 GOTO 0200
400 FOR T=U TO 30
410 PRINT AT U,U;"GAME OVER"
420 PRINT AT U,U;" "
430 NEXT T
440 REM HIGHEST SCORE ROUTINE
450 IF S<H1 THEN GOTO 0490
460 LET H2=H1
470 LET H1=S
480 GOTO 0510
490 IF S<H2 THEN GOTO 0510
500 LET H2=S
510 PRINT AT 5,16;H1;AT 6,16;H2
520 PRINT AT 9,U;"PRESS SPACE T
O STOP,","ANY OTHER KEY TO PLAY
AGAIN."
530 IF INKEY$="" THEN GOTO 0530
540 GOTO 0150

```

#### Quiz 1

```

10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 PRINT AT 8,9;"SCORE";F
T 9,9;"QUIZ 1";AT 10,9;"TIME"

```

```

50 PAUSE 130
60 DIM A$(8,13)
70 FOR H=5 TO U
80 LET A$(U)="51382DELETE"
90 LET A$(2)="267315"
100 LET A$(3)="15363"
110 LET A$(4)="148225SHIFT"
120 LET A$(5)="58361PRINT"
130 LET A$(6)="00000?"
140 LET A$(7)="000000"
150 LET A$(8)="000000ENTER"
160 IF H THEN GOSUB 0520
170 FOR P=U TO 5
180 CLS
190 PRINT TAB 8;"QUESTION ";P+(
5 AND H)
200 PRINT
210 REM PRINT CHOICES
220 FOR F=U TO 4
230 LET C=VAL A$(P,F)
240 PRINT F;" ";A$(C,6 TO )
250 NEXT F
260 LET C=CODE INKEY$-VAL "28"
270 IF C<U OR C>4 THEN GOTO 026
0
280 IF C=VAL A$(P,5) THEN GOTO
0470
290 IF NOT F THEN GOTO 0330
300 PRINT AT 12,5;"NOPE - TRY F
GAIN"
310 LET F=NOT U
320 GOTO 0260
330 PRINT AT 12,5;"ANSWER IS ";
A$(P,6 TO )
340 PRINT "      PRESS ""C"" TO C
O ON"
345 IF INKEY$<>"C" THEN GOTO 34
5
350 NEXT P
360 NEXT H
370 CLS
380 PRINT TAB 8;"YOUR SCORE"
390 PRINT AT 5,U;"FIRST TRY - "
;5
400 PRINT AT 7,U;"SECOND TRY -
";51
410 IF 5<8 THEN GOTO 0440
420 PRINT AT 12,5;"CONGRATULATI
ONS"
430 STOP
440 PRINT AT 12,0;"YOU NEED PRA
CTICE.", "STUDY A BIT MORE, THEN
TAKE", "THIS QUIZ AGAIN."
450 STOP
470 IF F THEN LET 5=5+U
480 IF NOT F THEN LET 51=51+U
490 PRINT AT 12,5;"THAT IS RIGH
T"
500 PAUSE 100
510 GOTO 0350

```

```

520 LET A$(1) = "16421RUN"
530 LET A$(2) = "82712LOAD"
540 LET A$(3) = "27363NEU"
550 LET A$(4) = "62144ENTER"
570 LET A$(6) = "00000SHIFT"
560 LET A$(7) = "00000PROGRAM"
590 LET A$(8) = "00000CASSETTE"
600 RETURN

```

## CHAPTER 2

### Game 2

```

10 LET U=PI/PI
15 LET F=VAL "10"
20 PRINT AT F-U,F,"";
; AT F,F;" GAME 2 "; AT F+U,F;"
30 PAUSE 150
40 CLS
50 PRINT " TYPE THE NUMBER
55 PRINT ",,,"0 RETURN TO MEN
U"
60 PRINT ",,,"1 DRAWING"
70 PRINT ",,,"2 FLASHING GRAPHI
CS"
80 PRINT ",,,"3 MAZE MAKER"
90 PRINT ",,,"4 STOP "
100 IF INKEY$="1" THEN GOTO 020
0
110 IF INKEY$="2" THEN GOTO 040
0
120 IF INKEY$="3" THEN GOTO 060
0
130 IF INKEY$="4" THEN STOP
140 GOTO 0100
200 CLS
210 PRINT AT VAL "14",NOT U;"US
E THE ARROWS ON KEYS 5, 6, 7 AN
D 8 TO DRAW."
220 PRINT ",,,"PRESS CLS TO CLEAR
THE SCREEN."
230 PRINT ",,,"PRESS ""0"" TO RET
URN "
235 PRINT ",,,"PRESS ""U"" TO ERA
SE, ""0"" TO DRAW"
240 LET X=VAL "31"
250 LET Y=VAL "28"
260 LET D$=INKEY$
265 IF D$="0" THEN LET F=U
265 IF D$="U" THEN LET F=NOT U
270 IF D$="U" THEN GOTO 0200
280 IF D$="0" THEN GOTO 0040
290 LET X=X-(U AND D$="5" AND X
>NOT U)+(U AND D$="8" AND X<63)
300 LET Y=Y+(U AND D$="7" AND Y
<43)-(U AND D$="6" AND Y>18)
310 IF F THEN PLOT X,Y
315 IF NOT F THEN UNPLOT X,Y

```

```

320 GOTO 0260
400 CLS
410 PRINT AT 11,8;"PRESS ""0""
TO RETURN "
420 DIM T(5)
425 UNPLOT NOT U,43
430 FOR X=U TO 5
440 LET T(X)=INT (RND*10)+U
450 NEXT X
460 FOR Y=U TO VAL "58"
470 FOR X=U TO 5
480 PRINT CHR$( T(X));
490 IF INKEY$=""0" THEN GOTO 004
0
500 NEXT X
510 NEXT Y
520 GOTO 0425
600 CLS
610 PRINT AT NOT U,U;"PRESS ""1
"" TO DO A NEW PATTERN."
620 PRINT AT U,U;"PRESS ""0"" T
O RETURN "
630 LET X=INT (RND*36)+U
640 LET Y=INT (RND*36)+U
650 LET T=INT (RND*2)+U
660 FOR F=U TO X STEP T
670 PLOT X,F
680 PLOT F,X
690 IF INKEY$=""0" THEN GOTO 004
0
700 IF INKEY$=""1" THEN GOTO 060
0
710 PLOT Y,F
720 PLOT F,Y
730 NEXT F
740 GOTO 0630

```

## Quiz 2

```

1 REM 00100
10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 PRINT AT 8,9;" ";F
T 9,9;" QUIZ 2 ";AT 10,9;" "
50 PAUSE 130
60 DIM A$(8,13)
70 FOR H=5 TO U
80 LET A$(U)="78163<="
90 LET A$(2)="25631R"
100 LET A$(3)="41383"
110 LET A$(4)="46131"
120 LET A$(5)="78454RUN"
130 LET A$(6)="00000INT"
140 LET A$(7)="00000NOT"
150 LET A$(8)="00000AT"
160 IF H THEN GOSUB 0520
170 FOR P=U TO 5
180 CLS

```



```

190 PRINT TAB 8;"QUESTION ";P+(
5 AND H)
200 PRINT
210 FOR F=U TO 4
220 LET C=VAL A$(P,F)
230 PRINT F;" ";A$(C,6 TO )
240 NEXT F
250 LET C=CODE INKEY$-28
260 IF C<U OR C>4 THEN GOTO 025
0
270 IF C=VAL A$(P,5) THEN GOTO
0470
280 IF NOT F THEN GOTO 0320
290 PRINT AT 12,5;"WRONG - TRY
AGAIN"
300 LET F=NOT U
310 GOTO 0250
320 PRINT AT 12,5;"ANSWER - ";A
$(P,6 TO )
330 PRINT TAB 5;"PRESS ""C"" TO
GO ON"
335 IF INKEY$<>"C" THEN GOTO 03
35
340 NEXT P
350 NEXT H
360 CLS
370 PRINT TAB 11;"YOUR SCORE"
380 PRINT AT 5,U;"FIRST TRY - "
;5
390 PRINT AT 7,U;"SECOND TRY -
";51
400 IF S<8 THEN GOTO 0430
410 PRINT AT 12,5;"CONGRATULATI
ONS"
420 STOP
430 PRINT AT 12,0;"YOU NEED PRA
CTICE."
440 PRINT "STUDY SOME MORE THEN
TAKE THIS","QUIZ AGAIN."
450 STOP
470 IF F THEN LET S=S+U
480 IF NOT F THEN LET S1=S1+U
490 PRINT AT 12,5;"YOU GOT IT
"
500 PAUSE 100
510 GOTO 0340
520 LET A$(U)="716225AVE"
530 LET A$(2)="52742PAUSE"
540 LET A$(3)="51834FUNCTION"
550 LET A$(4)="65483CLS"
560 LET A$(5)="57861GRAPHICS"
570 LET A$(6)="00000GOTO"
580 LET A$(7)="00000LOAD"
590 LET A$(8)="00000PRINT AT"
600 RETURN

```

## CHAPTER 3

### Game 3

5 REM ~~GAME 3~~

```

10 LET D=VAL "10"
20 LET U=D/D
40 LET TH=U
50 PRINT AT D-U,D; " ";
;AT D,D; "  GAME 3  ";AT D+U,D; "
60 PAUSE VAL "150"
70 CLS
80 PRINT TAB D+U; "RECTIONS"
90 PRINT "I WILL HIDE A TREASURE BOX SOME-WHERE ON THE SCREEN."
100 PRINT "TO FIND THE TREASURE YOU MUST", "GUESS FIRST THE ROW NUMBER AND ", "THEN THE COLUMN NUMBER."
110 PRINT "THE SIZE OF THE TREASURE DEPENDS ON HOW QUICKLY YOU SUCCEED."
120 PRINT "READY? (Y OR N)"
130 IF INKEY$ <> "Y" THEN GOTO 0130
30
140 DIM I$(U+U,INT PI)
150 LET I$(U)="ROW"
160 LET I$(U+U)="COL"
165 REM REDEFINE
170 CLS
175 LET P=25
180 LET C=VAL "22"
190 DIM R(U+U)
200 LET R(U)=INT (RND*C)
210 LET R(U+U)=INT (RND*32)
220 PRINT TAB VAL "5"; "TREASURE AT ";TAB D+D+U; ", ";TAB VAL "26"; ", ";
230 LET I=VAL "17"
235 REM REDEFINE
240 FOR G=U TO U+U
250 LET I=I+(VAL "5" AND G>U)
260 LET C=C+(D AND G>U)
270 LET P=P-U
280 FOR T=U TO D
290 PRINT AT NOT U,I; " ";AT NOT U,I;I$(G),,,
300 NEXT T
310 INPUT A
320 IF A<NOT U OR A>C-U THEN GOTO 0530
330 IF R(G)<A THEN LET S$=" < "
340 IF R(G)>A THEN LET S$=" > "
350 IF R(G)=A THEN LET S$=" = "
360 PRINT I$(G); " IS ";S$;A; "
370 IF A<>R(G) THEN GOTO 0270
380 PRINT AT NOT U,I; " ";AT NOT U,I+U+U;R(G)
390 NEXT G
400 PRINT AT U+U,NOT U; "YOU GOT IT."
410 FOR T=U TO VAL "50"

```

```

420 PRINT AT R(U),R(U+U);" ";AT
R(U),R(U+U);" "
430 NEXT T
440 IF P<U THEN LET P=U
450 LET T=P*P+INT (P*(RND+.5))*
2)
460 IF T>TH THEN LET TH=T
470 CLS
480 PRINT "THE TREASURE WAS $";
T
490 PRINT ,,,, "HIGHEST SCORE $"
;TH
500 PRINT AT D+D,U;"PRESS ""Y""
TO PLAY AGAIN"," SCREEN TO STOP
"
510 GOTO 0130
530 PRINT AT D,U;"OFF THE SCREE
N (0 TO ";C-U;")"
540 PAUSE D*D
550 PRINT AT D,U;"
(
560 GOTO 0270

```

### Quiz 3

```

5 REM QUIZ 3
6 REM
10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 LET T=U+U
50 LET Q=VAL "4"
60 LET U=Q+U
70 LET E=U+U
80 LET R=E+U
90 LET I=R+U
100 PRINT AT I,I+U;" "
;AT I+U,I+U;" QUIZ 3 ";AT 10,E
;" "
110 PAUSE VAL "150"
120 DIM A$(I,R)
130 FOR H=S TO T
140 LET A$(U)="12681*"
150 LET A$(T)="14233/"
160 LET A$(INT PI)="53742>"
170 LET A$(Q)="72644<="
180 LET A$(U)="00000>="
190 LET A$(E)="00000<>"
200 LET A$(R)="00000<"
210 LET A$(I)="00000X"
220 IF H=U THEN GOSUB 0590
230 IF H=T THEN GOSUB 0680
240 FOR P=U TO Q
250 CLS
260 PRINT TAB I;"QUESTION ";P+(
0*H AND H)
270 PRINT
280 FOR F=U TO Q
290 LET C=VAL A$(P,F)
300 PRINT F;" ";A$(C,E TO )

```

```

310 NEXT F
320 LET F=NOT U
330 LET C=CODE INKEY$-VAL "28"
340 IF C>0 OR C<U THEN GOTO 033
0
350 IF C=VAL A$(P,W) THEN GOTO
0540
360 IF F THEN GOTO 0400
370 PRINT AT E+E,U;"NOPE - TRY
AGAIN"
380 LET F=U
390 GOTO 0330
400 PRINT AT E+E,U;"ANSWER IS "
;A$(P,E TO );"
410 PRINT TAB U;"PRESS ""C"" TO
CONT "
415 IF INKEY$<>"C" THEN GOTO 41
5
420 NEXT P
430 NEXT H
440 CLS
450 PRINT TAB E+W;"YOUR SCORE"
460 PRINT AT W,U;"FIRST TRY - "
;S
470 PRINT AT R,U;"SECOND TRY -
";S1
480 IF S<I+U THEN GOTO 0510
490 PRINT AT E+E,U;"GOOD WORK.
ON TO CHAPTER 4"
500 STOP
510 PRINT AT E+E,NOT U;"YOU NEE
D PRACTICE.", "STUDY SOME MORE, T
HEN TRY AGAIN"
520 STOP
540 IF F THEN LET S1=S1+U
550 IF NOT F THEN LET S=S+U
560 PRINT AT E+E,U;"YOU GOT IT,
KID "
570 PAUSE VAL "100"
580 GOTO 0420
590 LET A$(U)="21642 LET "
600 LET A$(T)="62752 FAST "
610 LET A$(INT PI)="87363 GOSUE
"
620 LET A$(Q)="31844 RETURN "
630 LET A$(W)="00000 SAVE "
640 LET A$(E)="00000 SLOW "
650 LET A$(R)="00000 CLS "
660 LET A$(I)="00000 GOTO "
670 RETURN
680 LET A$(U)="145614"
690 LET A$(T)="7382440"
700 LET A$(INT PI)="7263470"
710 LET A$(Q)="214538"
720 LET A$(W)="000003"
730 LET A$(E)="000006"
740 LET A$(R)="0000030"
750 LET A$(I)="0000060"
760 RETURN

```

## CHAPTER 4

### Game 4

```

10 LET U=1
20 LET D=10
30 LET TW=U+U
40 LET F=D+D
50 LET B$="TTTT"
60 PRINT AT D-U,D;" ";AT D,D;"  GAME 4  ";AT D+U,D;" "
70 FOR T=1 TO 100
80 NEXT T
90 CLS
100 LET A$=" STOP A SLOW D FAST
   F NEW A SCROLL B TO 4 FOR F GOT
   O G GOSUB H RETURN Y LET L LIST
   K NEXT N PRINT P RUN R CLS U"
110 PRINT AT D*TW+U,NOT U;" "
120 FOR G=1 TO 25
130 PRINT AT U,D;"PRESS ""1"" T
   O ";AT TW,D;"GO ON."
140 LET R=INT (RND*VAL "16")+U
150 IF INKEY$<>"1" THEN GOTO 01
50
160 PRINT AT U,D;"
   "
170 PRINT AT TW,D;A$(R*TW-U);"
   "
180 FOR T=1 TO 100
190 IF INKEY$=A$(R*TW) THEN GOT
   O 0240
200 NEXT T
210 NEXT G
220 PRINT AT TW,D;"BETTER LUCK
   NEXT TIME"
230 STOP
240 IF F<VAL "6" THEN LET B$="I
   T"
250 IF F=TW THEN LET B$="T"
260 IF F<TW THEN LET B$=" "
270 PRINT AT F,U;B$
280 IF F=NOT U THEN GOTO 0310
290 LET F=F-U
300 GOTO 0210
310 FOR T=0 TO 0 STEP -U
320 PRINT AT TW,D;T;" "
330 FOR G=U TO D
340 NEXT G
350 NEXT T
360 PRINT AT TW,D;"LOOK OUT SEE
   ME"
370 PRINT AT NOT U,TW;" ";AT N
   OT U,TW;" "
380 PRINT AT U,TW;" ";AT U,TW;
   "

```

```

390 FOR F=TW TO VAL "21"
400 PRINT AT F,U;"██████";AT F,U
;
410 NEXT F
420 PRINT AT F-U,U;"███. "

```

#### Quiz 4

```

10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 LET T=U+U
50 LET Q=VAL "4"
60 LET W=Q+U
70 LET E=U+U
80 LET R=E+U
100 PRINT AT R,R+T;"██████████"
;AT R+U,R+T;"  QUIZ 4  ";AT W+Q,
W+Q;"██████████"
110 PAUSE VAL "150"
120 DIM A$(R,R+E)
130 FOR H=5 TO T
140 LET A$(U)="1461 LIST "
150 LET A$(T)="6723 CONT "
160 LET A$(INT PI)="7362 10 PRI
NT "
170 LET A$(Q)="1452 LIST 150"
180 LET A$(W)="0000 LIST -150"
190 LET A$(E)="0000 ENTER"
200 LET A$(R)="0000 CLS "
220 IF H=U THEN GOSUB 0590
230 IF H=T THEN GOSUB 0680
240 FOR P=U TO Q
250 CLS
260 PRINT TAB R;"QUESTION ";P+(
Q-H AND H)
270 PRINT
280 FOR F=U TO INT PI
290 LET C=VAL A$(P,F)
300 PRINT F;" ";A$(C,W TO )
310 NEXT F
330 LET C=CODE INKEY$-VAL "28"
340 IF C>INT PI OR C<U THEN GOT
O 0330
350 IF C=VAL A$(P,Q) THEN GOTO
0540
360 IF NOT F THEN GOTO 0400
370 PRINT AT E+E,W;"NOPE - TRY
AGAIN"
380 LET F=NOT U
390 GOTO 0330
400 PRINT AT E+E,W;"ANSWER IS "
;A$(P,W TO );"
410 PRINT TAB W;"PRESS ""C"" TO
GO ON"
415 IF INKEY$<>"C" THEN GOTO 41
5
420 NEXT P
430 NEXT H
440 CLS

```

```

450 PRINT TAB E+U;"YOUR SCORE"
460 PRINT AT U,U;"FIRST TRY - "
;S
470 PRINT AT R,U;"SECOND TRY - "
;S1
480 IF S<R+T THEN GOTO 0510
490 PRINT AT E+E,U;"GOOD WORK.
ON TO CHAPTER 4"
500 STOP
510 PRINT AT E+E,NOT U;"YOU NEE
D PRACTICE."
520 PRINT "STUDY SOME MORE, THE
N TRY AGAIN"
530 STOP
540 IF NOT F THEN LET S1=S1+U
550 IF F THEN LET S=S+U
560 PRINT AT E+E,U;"THAT IS THE
ONE "
570 PAUSE VAL "100"
580 GOTO 0420
590 LET A$(U)="1251 PRINT ""*""
;""*""
600 LET A$(T)="1523 PRINT ""*""
;""*""
610 LET A$(INT PI)="5372 SCROLL
30"
620 LET A$(Q)="6343 SCROLL
30"
630 LET A$(W)="0000 PRINT AT 8,
9;""*""
640 LET A$(E)="0000 CONT "
650 LET A$(R)="0000 PAUSE 30"
670 RETURN
680 LET A$(U)="3152FOR"
690 LET A$(T)="2351TO"
700 LET A$(INT PI)="1322NEXT"
710 LET A$(Q)="6743A+B"
720 LET A$(W)="0000LET"
730 LET A$(E)="0000235"
740 LET A$(R)="0000200 + 35"
760 RETURN

```

## CHAPTER 5

### Game 5

```

10 REM GAME 5
20 LET A=VAL "9"
30 PRINT AT VAL "8",A;"
";AT A,A;" GAME 5 ";AT A+A,
A,A;"
40 LET S=VAL "5"
50 PAUSE VAL "150"
60 DIM A$(3,5)
70 LET A$(1)="LIKES"
80 LET A$(2)="LOVES"
90 LET A$(3)="HATES"
100 CLS
110 REM END OF GAME 5
120 PRINT " INPUT 5 GIRLS NAMES
.", " THEN INPUT 5 BOYS NAMES."

```

```

130 PRINT "..."EDIT
140 DIM N$(VAL "10",A)
150 FOR N=1 TO 10
160 PRINT N;" ";
170 INPUT N$(N)
180 PRINT N$(N)
190 IF N=5 THEN PRINT "..."EDIT
200 NEXT N
210 REM EDIT EDIT
220 CLS
230 LET C=NOT 5
240 PRINT AT VAL "18",NOT 5;"PRESS " "1" " TO EDIT NAMES."
250 PRINT "PRESS ANY KEY TO GO ON"
260 UNPLOT NOT 5,VAL "42"
270 REM CHANGE RANDOM MESSAGE
280 LET A=INT (RND*3)+1
290 LET N=INT (RND*5)+1
300 LET B=INT (RND*5)+VAL "6"
310 LET C=C+1
320 PRINT
330 REM PRINT A NEW NOTE
335 REM BOY OR GIRL FIRST?
340 IF RND>.5 THEN GOTO 0370
350 PRINT N$(N);TAB 5+5;A$(A);TAB VAL "19";N$(B)
360 GOTO 380
370 PRINT N$(B);TAB 5+5;A$(A);TAB VAL "19";N$(N)
380 PRINT
390 IF INKEY$="" THEN GOTO 390
400 IF INKEY$="1" THEN GOSUB 440
0
410 IF C=VAL "6" THEN GOTO 220
420 GOTO 280
430 REM EDITING SUBROUTINE
440 CLS
450 PRINT "YOUR LIST:",,,,
460 FOR N=5/5 TO 5+5
470 PRINT N;" ";N$(N);
480 NEXT N
490 PRINT AT VAL "10",NOT 5;"NUMBER OF NAME TO CHANGE?"
500 INPUT N
510 IF N>5+5 THEN GOTO 490
520 PRINT AT VAL "10",NOT 5;"NOW ENTER THE NEW NAME. "
530 INPUT N$(N)
540 PRINT AT PI/PI+INT N/2,VAL "19"-(VAL "32"+(N/2-INT (N/2)));N$(N)
550 PRINT AT VAL "10",NOT 5;"ANY MORE CHANGES?(Y OR N)"
560 IF INKEY$="Y" THEN GOTO 490
570 IF INKEY$<>"N" THEN GOTO 560
0
580 CLS
590 GOTO 230

```



# Quiz 5

```

10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 LET T=U+U
50 LET Q=VAL "4"
60 LET U=Q+U
70 LET E=U+U
80 LET R=E+U
90 PRINT AT R,R+T; " "
: AT R+U,R+T; " QUIZ 5 "; AT U+Q,
U+Q; " "
100 PAUSE VAL "150"
110 DIM A$(R,R+U)
120 FOR H=5 TO T
130 LET A$(U)="5122INPUT"
140 LET A$(T)="2351REM"
150 LET A$(INT PI)="6133BREAK"
160 LET A$(Q)="7143FOR... TO "
170 LET A$(U)="0000ENTER"
180 LET A$(E)="0000STOP"
190 LET A$(R)="0000LET"
200 IF H=U THEN GOSUB 0570
210 IF H=T THEN GOSUB 0650
220 FOR P=U TO Q
230 CLS
240 PRINT TAB R;"QUESTION ";P+(
Q-H AND H)
250 PRINT
260 FOR F=U TO INT PI
270 LET C=VAL A$(P,F)
280 PRINT F;" ";A$(C,U TO )
290 NEXT F
300 LET C=CODE INKEY$-VAL "28"
310 IF C>INT PI OR C<U THEN GOT
O 0300
320 IF C=VAL A$(P,Q) THEN GOTO
0520
330 IF NOT F THEN GOTO 0370
340 PRINT AT E+E,U;"WRONG- TRY
AGAIN"
350 LET F=NOT U
360 GOTO 0300
370 PRINT AT E+E,U;"ANSWER IS "
;A$(P,U TO );" "
380 PRINT TAB U;"PRESS ""C"" TO
GO ON"
390 IF INKEY$<>"C" THEN GOTO 39
0
400 NEXT P
410 NEXT H
420 CLS
430 PRINT TAB E+U;"YOUR SCORE"
440 PRINT AT U,U;"FIRST TRY - "
;S
450 PRINT AT R,U;"SECOND TRY -
";S1
460 IF S<R+T THEN GOTO 0490
470 PRINT AT E+E,U;"CONGRATULAT

```

IONS." ON TO CHAPTER 6"

```
480 STOP
490 PRINT AT E+E,NOT U;"YOU NEE
D PRACTICE."
500 PRINT "STUDY SOME MORE, THE
N TRY AGAIN"
510 STOP
520 IF NOT F THEN LET S1=S1+U
530 IF F THEN LET S=S+U
540 PRINT AT E+E,U;"SMART KID

550 PAUSE VAL "100"
560 GOTO 0400
570 LET A$(U)="2313INT (RND*9) +
1"
580 LET A$(T)="5212INT (RND*9)"
590 LET A$(INT PI)="3521(RND*9)
+1"
600 LET A$(0)="6743 DIM B$(3,8)
610 LET A$(U)="0000(RND*4)+1"
620 LET A$(E)="0000 DIM B$"
630 LET A$(R)="0000 LET B$=""3"
640 RETURN
650 LET A$(U)="2152INPUT A$"
660 LET A$(T)="2151INPUT A"
670 LET A$(INT PI)="36711"
680 LET A$(0)="76436"
690 LET A$(U)="0000 INPUT A$(1)
700 LET A$(E)="00002"
710 LET A$(R)="00003"
720 RETURN
```

## CHAPTER 6

### Game 6

```
10 REM GAME 6
20 LET U=PI/PI
30 LET SC=NOT U
40 LET S1=SC
50 LET S2=S1
60 LET D=10
70 DIM U$(10,10)
80 LET U$(1)="PROGRAMMER"
90 LET U$(2)="COMPUTER"
100 LET U$(3)="INPUT"
110 LET U$(4)="BASIC"
120 LET U$(5)="GRAPHICS"
130 LET U$(6)="FUNCTIONS"
140 LET U$(7)="PRINT"
150 LET U$(8)="CLARA"
160 LET U$(9)="LIST"
170 LET U$(10)="SCROLL"
180 PRINT AT D-U,D;"GAME 6"
:AT D,D;"GAME 6";AT D+U,D;"GAME 6"
190 PAUSE 100
```

```

200 CLS
210 REM CHOOSE WORD CHECK FOR
210
220 LET U=INT (RND*D)+U
230 FOR V=U TO D
240 IF W$(U,V)=" " THEN GOTO 02
60
250 NEXT V
260 LET V=V-U
270 DIM U(V)
280 FOR L=U TO V
290 LET U(L)=NOT U
300 NEXT L
305 FAST
310 PRINT TAB D;"UNSCRAMBLE IT"
320 UNPLOT NOT U,32
325 REM CHOOSE WORD
330 FOR L=U TO V
340 LET R=INT (RND*V)+U
350 IF U(R) THEN GOTO 0340
360 PRINT W$(U,R);
370 LET U(R)=U
380 NEXT L
385 SLOW
390 LET SC=SC+U
395 REM CHOOSE INPUT
400 INPUT Z$
410 IF Z$=W$(U) (U TO V) THEN GO
TO 0510
420 PRINT ,,,, " ";Z$,"NOPE"
430 LET 0=U
440 IF LEN Z$<V THEN LET 0=LEN
Z$
445 REM EIGHT LETTERS
450 FOR T=U TO 0
460 IF Z$(T)=W$(U,T) THEN PRINT
TAB T;" ";
470 NEXT T
480 IF NOT L THEN GOTO 0620
490 LET L=NOT L
500 GOTO 0400
510 PRINT ,,,, "GET IT"
520 LET S1=S1+(U AND L)
530 LET S2=S2+(U AND NOT L)
540 PRINT ,,,, "U" FOR ANOTHER
WORD", "" "A" TO STOP "
550 IF INKEY$="U" THEN GOTO 020
0
550 IF INKEY$<>"A" THEN GOTO 05
50
570 CLS
575 REM SUB GAME
580 PRINT "FIRST GUESS",S1
590 PRINT "SECOND GUESS",S2
600 PRINT "MISSES",SC-(S1+S2)
610 STOP
620 PRINT ,,"THE WORD WAS ";W$(
U)
630 GOTO 0540

```

## Quiz 6

```

10 LET S=NOT PI
20 LET S1=S
30 LET U=PI/PI
40 LET T=U+U
50 LET Q=VAL "4"
60 LET W=Q+U
70 LET E=U+U
80 LET R=E+U
90 PRINT AT R,R+T;"██████████"
; AT R+U,R+T;"  QUIZ 6  "; AT W+Q,
W+Q;"██████████"
100 PAUSE VAL "150"
110 DIM A$(R,R+E)
120 FOR H=S TO T
130 LET A$(U)="3172EDIT"
140 LET A$(T)="2541██"
150 LET A$(INT PI)="4362DOWN AR
R."
160 LET A$(Q)="3452LEFT ARR."
170 LET A$(W)="0000UP ARR."
180 LET A$(E)="0000AT, ARR."
190 LET A$(R)="0000NEW"
200 IF H=U THEN GOSUB 570
210 IF H=T THEN GOSUB 650
220 FOR P=U TO Q
230 CLS
240 PRINT TAB R;"QUESTION ";P+(
Q-H AND H)
250 PRINT
260 FOR F=U TO INT PI
270 LET C=VAL A$(P,F)
280 PRINT F;" ";A$(C,W TO )
290 NEXT F
300 LET C=CODE INKEY$-VAL "28"
310 IF C>INT PI OR C<U THEN GOT
O 0300
320 IF C=VAL A$(P,Q) THEN GOTO
0520
330 IF NOT F THEN GOTO 370
340 PRINT AT E+E,W;"NOPE - TRY
AGAIN"
350 LET F=NOT U
360 GOTO 300
370 PRINT AT E+E,W;"ANSWER IS "
; A$(P,U TO );"
380 PRINT TAB W;"PRESS 'C'" TO
GO ON"
390 IF INKEY$(<"C" THEN GOTO 39
0
400 NEXT P
410 NEXT H
420 CLS
430 PRINT TAB E+W;"YOUR SCORE"
440 PRINT AT W,U;"FIRST TRY - "
; 5
450 PRINT AT R,U;"SECOND TRY -
"; 51
460 IF S<R+T THEN GOTO 490

```

```

470 PRINT AT E+E,NOT U;"YOU GOT
THROUGH THE LOUERS BOOK."
480 STOP
490 PRINT AT E+E,NOT U;"YOU NEE
D PRACTICE."
500 PRINT "STUDY SOME MORE, THE
N TRY AGAIN"
510 STOP
520 IF NOT F THEN LET S1=S1+U
530 IF F THEN LET S=S+U
540 PRINT AT E+E,U;"RIGHT YOU F
RE
550 PAUSE VAL "100"
560 GOTO 400
570 LET A$(U)="3172 PRINT ""DUP
MY""""
580 LET A$(T)="5323INT (RND*10)
+1"
590 LET A$(INT PI)="6342 INPUT
I"
600 LET A$(Q)="4751 IF I=A THEN
GOTO 90"
610 LET A$(W)="0000 INT (RND*10
)"
620 LET A$(E)="0000 INPUT I$"
630 LET A$(R)="0000 PRINT ""RIG
HT""""
640 RETURN
650 LET A$(U)="5313 REM LOUERS"
660 LET A$(T)="3523 GOTO 30"
670 LET A$(INT PI)="3211 STOP "
680 LET A$(Q)="467130"
690 LET A$(W)="0000 GOTO 50"
700 LET A$(E)="000040"
710 LET A$(R)="000060"
720 RETURN

```

## PROGRAM NOTES

### Game 1

- 10 GAME 1
- 70 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q
- 100 DIRECTIONS
- 300 WRONG
- 340 TIME
- 410 GAME OVER

### Quiz 1

- 40 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q

210 PRINT CHOICES  
380 YOUR SCORE

## Game 2

- 20 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q  
50 TYPE THE NUMBER YOU WISH

## QUIZ 2

- 40 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q  
100 Graphics R  
110 Inverse R

## Game 3

- 50 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q  
80 DIRECTIONS  
420 Inverse +  
500 BREAK

## Quiz 3

- 100 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q

## Game 4

- 50 Graphics E four times  
60 Graphics E, Graphics 7 eight times, Graphics R;

- Graphics 5, Graphics 8; Graphics W, Graphics 6 eight times, Graphics Q
- 110 Graphics Q, Inverse space five times, Graphics 4
- 220 **BETTER LUCK NEXT TIME**
- 240 Graphics 8, Graphics R three times
- 250 Graphics Q, Graphics R, Graphics 4
- 260 Graphics 5
- 360 **LOOK OUT BELOW**
- 370 Graphics 6
- 380 Graphics 6, Graphics T
- 400 Graphics T four times, Graphics 4
- 420 Graphics Q, Graphics 4, Inverse space, Graphics Q, Graphics 3

#### **Quiz 4**

- 100 Graphics E, Graphics 7 eight times, Graphics R; Graphics 5, Graphics 8; Graphics W, Graphics 6 eight times, Graphics Q

#### **Game 5**

- 30 Graphics E, Graphics 7 eight times, Graphics R; Graphics 5, Graphics 8; Graphics W, Graphics 6 eight times, Graphics Q
- 110 **INPUT NAMES**
- 130 **LADIES**
- 190 **GENTS**
- 210 **GOTO EDIT?**
- 270 **CHOOSE RANDOM MESSAGE**
- 330 **PRINT LOVE NOTE**
- 430 **EDITING SUBROUTINE**

#### **Quiz 5**

- 90 Graphics E, Graphics 7 eight times, Graphics R; Graphics 5, Graphics 8; Graphics W, Graphics 6 eight times, Graphics Q

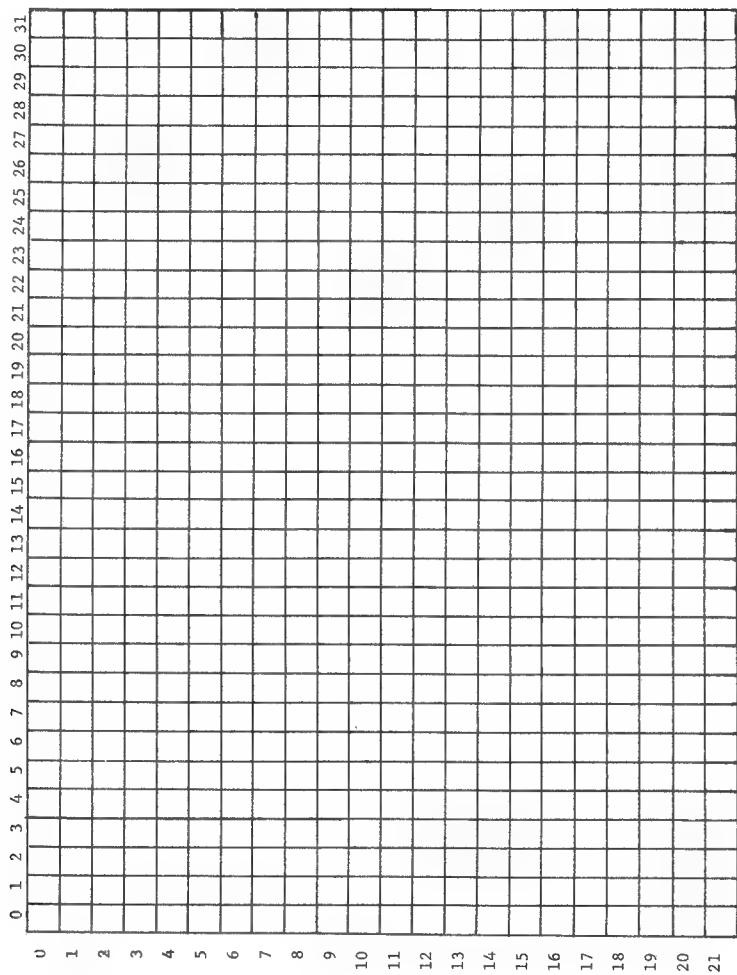
## Game 6

- 180 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q
- 210 CHOOSE WORD, CHECK FOR END
- 325 SCRAMBLE WORD
- 395 CHECK INPUT
- 420 NOPE
- 445 RIGHT LETTERS
- 510 GOT IT
- 575 END GAME

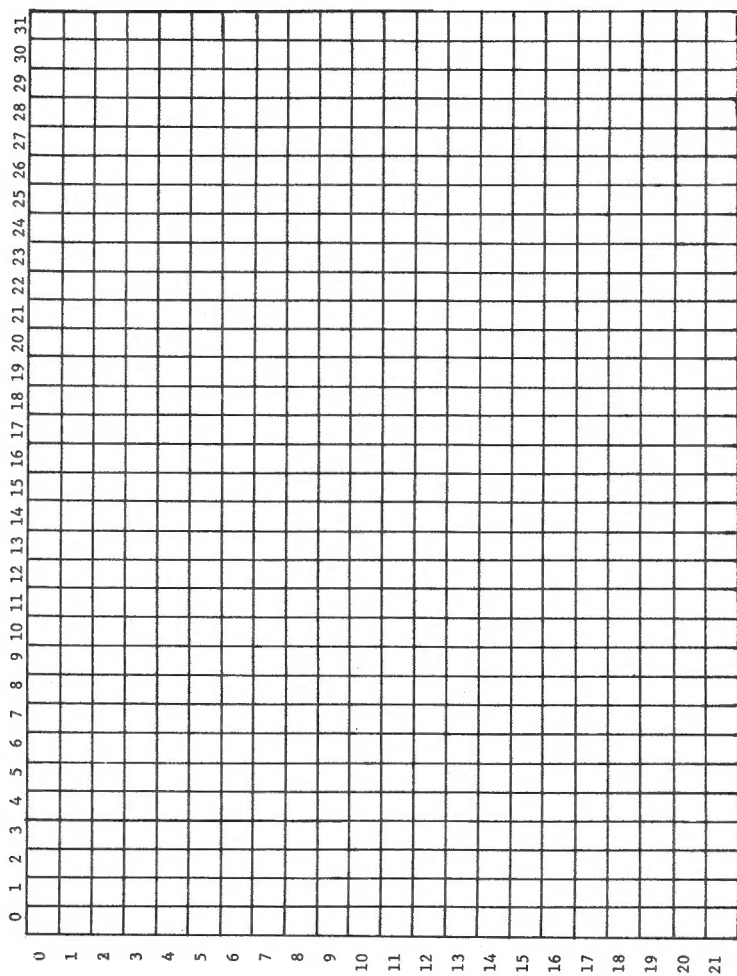
## Quiz 6

- 90 Graphics E, Graphics 7 eight times, Graphics R;  
Graphics 5, Graphics 8; Graphics W, Graphics 6 eight  
times, Graphics Q
- 140 Inverse greater than
- 470 WHOLE
- 650 LOVERS









# Index

---

- A**  
Addition, 44-45  
Array, 94
- B**  
BASIC, 2  
BREAK, 39-40  
Bugs, 13
- C**  
Comma, 76-78  
CONT, 73-74  
COS, 47  
Cosine, 47  
Cursor, 115
- D**  
DELETE, 6, 7  
DIM, 95-96  
Dimensions, 95-96  
Division, 45-46
- E**  
EDIT, 113-115  
ENTER, 4, 7  
E+, 51-52  
Equations, 84
- F**  
FAST, 48-49  
Flowchart, 61-63  
FOR . . . NEXT, 66-68  
FOR . . . TO, 64-69  
FUNCTION, 35  
Functions, 37
- G**  
GOSUB, 52-54  
GOTO, 38-40  
GRAPHICS, 25-43  
Graphics pad, 32-33
- I**  
IF . . . THEN, 107-109  
INKEY\$, 111-113  
Input, 82-104  
INPUT, 82-83  
INT, 98-99  
Inverse printing, 26
- K**  
Keyword, 22-23
- L**  
LET, 49-51  
Letter, 23-24  
Line numbers, 10-11  
LIST 71-72  
LN, 47  
Logarithms, 46-47  
Loops, 64-69
- M**  
Mathematics, 44-59

Menu driven, 28  
Multiplication, 45-46

## N

N\$ (N string), 86-88  
NEW, 11-12  
NEXT, 64-69  
N string, 86-88

## P

PAUSE, 39-40  
PRINT 2, 5  
PRINT AT, 34-35  
Printing, inverse, 26  
Program, 10-11

## Q

Quotation marks, 5

## R

Random numbers, 96-98  
REM, 92-93  
Remark, 92-93  
RETURN, 53-54  
RND, 98-100  
RUN, 10, 11

## S

SAVE, 32-34  
SCROLL, 67-68  
Semicolon, 75  
SHIFT, 3, 7  
SIN, 47  
Sine, 47  
SLOW, 48-49  
SPACE, 5  
Speech lessons, 1-19  
SQR, 46  
Square roots, 46  
STEP, 101-102  
STOP, 53-54  
String variables, 86-88  
Subroutines, 52-54  
Subtraction, 44-45  
Syntax error, 8

## T

TAN, 47  
Tangent, 47  
Tape recorder, using, 14-15, 130-131

## V

Variables, 49-51, 55-57  
Variables, string, 86-88

